# Université Paris-Sud

Ecole Doctorale D'Informatique Paris-Sud
Laboratoire : INRIA, Saclay, Ile de France ; CNRS-LIMSI, Orsay, France

Discipline : Informatique

## Thèse de doctorat

Soutenue le 17/11/2014 par


# Charles Perin


# Direct Manipulation for Information Visualization

| | | |
|---|---|---|
| **Directeur de thèse** : | M. Jean-Daniel Fekete | Directeur de Recherche (INRIA Saclay, France) |
| **Co-directeur de thèse** : | M. Frédéric Vernier | Maître de Conférences (CNRS-LIMSI, Université Paris-Sud, France) |

| | | |
|---|---|---|
| Composition du jury : | | |
| Président du jury : | Michel Beaudouin-Lafon | Professeur (laboratoire de Recherche en informatique, Université Paris-Sud, France) |
| Rapporteurs : | Michael McGuffin | Professeur (École de Technologie Supérieure, Canada) |
| | Ji Soo Yi | Professeur (Purdue University, USA) |
| Examinateurs : | James Eagan | Maître de Conférences (Institut Mines Telecom, France) |
| | Bongshin Lee | Chercheuse (Microsoft Research Redmond, USA) |

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES



# ACRONYMS

| | |
|---|---|
| HCI | Human Computer Interaction |
| Infovis | Information Visualization |
| WIMP | Windows, Icons, Menus, Pointer |
| GUI | Graphical User Interface |
| HTML | Hypertext Markup Language |
| RLC | Reduced Line Charts |
| HG | Horizon Graphs |
| IHG | Interactive Horizon Graphs |
| LSV | Large scale and Small scale Variations dataset |

# GLOSSARY

The definitions provided in this (non-exhaustive) glossary are of stipulative nature. These apply to this thesis and I do not claim they generalize.

**congruence** I use the definition of congruence for graphics: *"The structure and content of the external representation should correspond to the desired structure and content of the internal representation"* [Tversky et al., 2002].

**cognitively congruent** I use cognitive congruence to qualify the similarity between the intent of the user and the action she performs.

**congruent interaction** I call a congruent interaction an interaction whose action, reaction and feedback are cognitively congruent, and that provides a natural mapping between the action and the task performed through this action.

**congruent question** I use the following definition of a congruent question: the level of syntactic congruence for the coupling of visual representation–task is the *"replaceability of the data-related terms by perceptual terms"* [Boy et al., 2014].

**congruent** *See* congruence.

**discoverability** I consider discoverability as the starting point of learning, and assuming that interactions can be discovered either by exploring the interface, either by chance. Discoverability is related to the level of engagement of the user.

**discoverable** refers to an interaction or an interface that users can discover and learn by themselves. *See* discoverability.

**engagement** refers to the feeling of direct engagement, of first-personness, with the interface [Laurel, 1986]

**engage** *See* engaged.

**engaged** describes users who are eager to explore the visualization and discover the available interaction techniques*See* engagement.

**engaging** a visually engaging interface or visualization invites experimentation and use Dix and Ellis. *See* engagement.

**external consistency** of an interface refers to the similarity with other interface designs familiar to a user [Grudin, 1989], improving transfer of training.

**internal consistency** of an interface refers to the consistency of a design with itself [Grudin, 1989], improving learning, ease of use, and perceived quality of the interface.

**object of interest** refers to the (representation of an) object on which an action has an effect. Identifying the object of interest is an essential step when designing interaction.

**versatile** *See* versatility.
**versatility** I call an interaction or an instrument *versatile* if it is designed to perform a wide range of tasks. Versatility may conflict with congruence.

# ABSTRACT

There is a tremendous effort from the Information Visualization (Infovis) community to design novel, more efficient or more specialized desktop visualization techniques. While visual representations and interactions are combined to create these visualizations, less effort is invested in the design of new interaction techniques for Infovis. In this thesis, I focus on interaction for Infovis and explore how to improve existing visualization techniques through efficient yet simple interactions. To become more efficient, the interaction techniques should reach beyond the standard widgets and Window/Icon/Menu/Pointer (WIMP) user interfaces.

In this thesis, I argue that the design of novel interactions for visualization should be based on the direct manipulation paradigm and the instrumental interaction framework, and take inspiration from advanced interactions investigated in HCI research but not well exploited yet in Infovis. I extract from the HCI literature a large set of principles, benefits and challenges of direct manipulation interfaces that have been proposed and discussed over the last 30 years, that I classify into three high level overlapping categories: Learning, Ease of use, and Seamless and fluid interaction. Then I describe and evaluate several exemplar visualization techniques according to these criteria. I describe multiple projects I have designed based on these principles and benefits, to tackle direct manipulation challenges, illustrating how opportunistic interactions can empower visualizations. I explore design implications raised by novel interaction techniques, such as the tradeoff between cognitive congruence (the natural mapping between user's intent and action) and versatility (of the interaction techniques), the problem of engaging interaction (how to make the user engaged and willing to explore the visualization), and the benefits of seamless, fluid interaction. Finally, I provide design guidelines and perspectives, addressing the grand challenge of building or consolidating the theory of interaction for Infovis.

RÉSUMÉ

La communauté de la Visualisation d'Information (Infovis) accorde une importance primordiale à la conception de techniques de visualisation nouvelles, efficaces, ou spécialisées. Alors qu'une technique de visualisation est composée à la fois de techniques de représentation et de techniques d'interaction, la conception de nouvelles techniques d'interaction pour l'Infovis passe souvent au second plan. Dans cette thèse, centrée sur l'interaction en Infovis, j'explore la conception de nouvelles techniques d'interaction afin de rendre des techniques de visualisation existantes plus efficaces, plus adaptées aux tâches utilisateur, et plus engageantes. Afin que ces techniques d'interaction soient efficaces, il est nécessaire d'abandonner les outils interactifs (widgets) standards et proposer des interfaces utilisateur allant au-delà des interfaces à fenêtres, icônes, menus et pointeur connues sous le nom d'interfaces WIMP (Window/Icon/Menu/Pointer).

Dans cette thèse, je soutiens que la conception de nouvelles techniques d'interaction pour la visualisation devraient être basée sur le paradigme de la manipulation directe et sur le modèle de l'interaction instrumentale, et s'inspirer de paradigmes d'interaction établis en Interaction Homme-Machine (IHM) mais trop peu connus et reconnus en Infovis. En me basant sur plusieurs projets que j'ai menés au court de ma thèse, je démontre que la conception opportuniste d'interactions nouvelles peut rendre des techniques de visualisation plus efficaces. Ces différents projets soulèvent des problèmes de conception des techniques d'interaction, tels que le compromis entre la congruence cognitive d'une technique d'interaction et sa généricité, le problème de la conception d'interactions engageant l'utilisateur, et les mérites de l'interaction fluide et ininterrompue. Enfin, je propose un ensemble de règles dérivées des différents projets de cette thèse et je soumets des perspectives de travaux futurs, afin de contribuer au grand défi d'établir une théorie de l'interaction pour l'Infovis.

# PUBLICATIONS

Parts of this dissertation work have been published as full papers in international conferences, journals and poster presentations. This section lists all refereed articles published during my PhD research. Related chapters in this thesis are indicated in parenthesis.

### Papers in International Journals or Conferences

[1] **Charles Perin**, Pierre Dragicevic, and Jean-Daniel Fekete. Revisiting Bertin Matrices: New Interactions for Crafting Tabular Visualizations. IEEE Transactions on Visualization and Computer Graphics, 2014. IEEE. To appear. (Chapter 6)

[2] **Charles Perin**, Romain Vuillemot, and Jean-Daniel Fekete. À Table!: Improving Temporal Navigation in Soccer Ranking Tables. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '14, pages 887–896, Toronto, Canada, 2014. ACM. (Chapter 4)

[3] **Charles Perin**, Romain Vuillemot, and Jean-Daniel Fekete. SoccerStories: A Kick-off for Visual Soccer Analysis. IEEE Transactions on Visualization and Computer Graphics, 19(12):2506–2515, 2013. Infovis best paper honorable mention. IEEE.

[4] **Charles Perin**, Frédéric Vernier, and Jean-Daniel Fekete. Interactive Horizon Graphs: Improving the Compact Visualization of Multiple Time Series. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, pages 3217–3226, Atlanta, GA, USA, 2013. ACM. (Chapter 3)

### Papers at French Conferences

[5] **Charles Perin** and Pierre Dragicevic. Manipulation de Sliders Multiples par Franchissement. In Proceedings of the 26ième Conférence Francophone sur L'Interaction Homme-Machine, IHM '14, Lille, France, 2014. ACM. To appear. (Chapter 6)

[6] **Charles Perin**, Marc Christie, Frédéric Vernier, and Christophe Lino. CollaStar: Interaction Collaborative avec des Données Multidimensionnelles et Temporelles. In Proceedings of the 25ième Conférence Francophone sur L'Interaction Homme-Machine, IHM '13, pages 115:115–115:124, Bordeaux, France, 2013. ACM.

**Posters at International Conferences**


[7] Benjamin Bach, Pierre Dragicevic, Samuel Huron, Petra Isenberg, Yvonne Jansen, **Charles Perin**, André Spritzer, Romain Vuillemot, Wesley Willett, and Tobias Isenberg. Illustrative Data Graphics in 18th-19th Century Style: A Case Study. In IEEE Conference on Visualization - VIS '13, Atlanta, GA, United States, 2013. IEEE.

[8] Pierre Dragicevic, Benjamin Bach, Nicole Dufournaud, Samuel Huron, Petra Isenberg, Yvonne Jansen, **Charles Perin**, André Spritzer, Romain Vuillemot, Wesley Willett, and Tobias Isenberg. Visual Showcase: An Illustrative Data Graphic in an 18th-19th Century Style. In Visual Showcase at the Joint ACM/EG Symposium on Computational Aesthetics, Sketch-Based Interfaces and Modeling, and Non-Photorealistic Animation and Rendering (Expressive 2013), Anaheim, CA, United States, 2013.

[9] **Charles Perin**. CinemAviz. In VAST Challenge 1 (VIS '13), Atlanta, GA, USA, 2013.

[10] **Charles Perin**, Frédéric Vernier, and Jean-Daniel Fekete. Progressive Horizon Graphs: Improving Small Multiples Visualization of Time Series. In IEEE, editor, VisWeek 2012 Electronic Conference Proceedings, Seattle, WA, USA, 2012. IEEE. Infovis best poster honorable mention. (Chapter 3)


**Refereed Workshop Papers**


[11] **Charles Perin**. Expert Player Interface Design: Preserving the Flow. In Player Experience: Mixed Methods and Reporting Results (CHI '14). Toronto, Canada, 2014. ACM

[12] **Charles Perin**, Romain Vuillemot, and Jean-Daniel Fekete. Real-Time Crowdsourcing of Detailed Soccer Data. In What's the score? The 1st Workshop on Sports Data Visualization (VIS '13), Atlanta, GA, USA, 2013. IEEE.

[13] **Charles Perin** and Frédéric Vernier. R2S2: A Hybrid Technique to Visualize Sport Ranking Evolution. In What's the score? The 1st Workshop on Sports Data Visualization (VIS '13), Atlanta, GA, USA, 2013. IEEE. (Chapter 3)

# ACKNOWLEDGMENTS


As I expect this page to be the most read of this document, I will try not to forget anyone, although I am sure i will.

The first two people I want to strongly acknowledge are Jean-Daniel Fekete and Frédéric Vernier, who advised me during this 3-years journey. For offering me the opportunity to go into research, allowing me to realize what I love and want to do, as well as for trusting me—even my weird projects. I also particularly thank Pierre Dragicevic and Romain Vuillemot for working closely with me on one or several projects, during which I learned so much, from statistics to writing, from evaluations to design, resulting in the projects I am the most proud of and enjoyed the most. I hope they enjoyed it too, working as equal sibling with a more experienced researcher being certainly the most enjoyable for a PhD student.

I very thank the members of my jury: Michel Beaudouin-Lafon, James Eagan, Bongshin Lee, Michael McGuffin, and Ji Soo Yi, for accepting to review my thesis, their feedback, and attending my defense physically or virtually.

Then, I want to thank all people from the Aviz team, and I may change my mind later, but to the best of my knowledge, I do not see a better place for me to do a PhD. André Spritzer for helping me a lot with English writing, Yvonne Jansen and Pierre Dragicevic for the numerous fruitful discussions while smoking coffees in the cold, Benjamin Bach for being a quiet office partner and always available, Manfred Micaux and Kai Lawonn who did not stay around long enough, Wesley Willett and Lora Oehlberg for being such nice, knowledgeable, and enthusiastic people (and thanks Wes for this really nice bike tour to Normandy), Petra and Tobias Isenberg for being always available to answer questions, Nadia Boukhelifa, Anthi Dimara, Pascal Goffin, and Romain Di Vozzo, always interested when talking to them, Nicolas Heulot for the wine lessons, and Jeremy Boy, Mathieu Le Goc and Samuel huron for convincing me to go for $n \in [1, \infty]$ drinks, fruitful discussions, and making conferences so enjoyable.

I also thank all football players from LIMSI and InSitu—although it broke successively my left ankle, my right ankle, and one one of my ribs, some InSitu people I love chatting with and meeting during conferences (Olivier, Stéphane, Anastasia, Jeremie, and many others), and Fanny Chevalier for being so helpful for many things and trusting me.

Finally, I want to thank my family, especially my mother, for supporting me all these years, and last but not least apologize to both Céline and Dimanche, respectively my almost–wife and cat, for the difficult moments we went through due to the massive work and investments I put into this 3 years thesis.

# PREAMBLE

In 1953, master of hard science fiction Isaac Asimov writes about a futuristic device in his book *Second Foundation* [Asimov, 1953] (pp. 30–37), the *visiplate*. Thirty years before Information Visualization was established as a research field, Isaac Asimov imagined that visualizing data would be on the cutting edge of technology, thousands of years after the universe was explored. He envisioned that visualization would overcome mathematical equations for complex tasks involving massive amounts of data.

*The Lens was perhaps the newest feature of the interstellar cruisers of the day. Actually, it was a complicated calculating machine which could throw on a screen a reproduction of the night sky as seen from any given point of the Galaxy. [...] it was further capable of translating any given portion of the Galactic Field along any of the three spatial axes or to rotate any portion of the Field about a center. It was because of that, that the Lens had performed a near-revolution in interstellar travel. In the younger days of interstellar travel, the calculation of each "hop" through hyperspace meant any amount of work from a day to a week.*

His second anticipation was that operating such a system would not require any expert skills and no particular background.

*[...] It was the Lens that changed all that. For one thing it required only a single known star. For another, even a space tyro such as Channis could operate it.*

His last prediction was that interaction between the human and the device would be required.

*[...] And Channis pointed again, [...] and Channis' finger silently followed it down, to where it straggled to a halt, and then, as his finger continued moving onward, to a spot where one single star sparked lonesomely; and there his finger halted, for beyond that was blackness, unrelieved.*

*[...] and with careful fingers, Channis punched out the co-ordinates of Vincetori. He closed a relay, and the star field sprang to bright view. In it, too, a bright star was centered, but otherwise there seemed no relationship. He adjusted the Lens along the Z-Axis and expanded the Field to where the photometer showed both centered stars to be of equal brightness. Channis looked for a second star, sizably bright, on the visiplate and found one on the field screen to correspond. Slowly, he rotated the screen to similar angular deflection. [...] Again he rotated and another bright star was brought into position, and a third. [...] In the final step, the two fields overlapped and merged into a sea of not-quite-rightness. Most of the stars were close doubles. But the fine adjustment did not take long. The double stars melted together, one field remained, and the "Ship's Position" could now be read directly off the dials. The entire procedure had taken less than half an hour.*

Isaac Asimov imagined that visualization would be the ultimate tool for analyzing data, and he imagined visualization with advanced human interaction.

# 1 INTRODUCTION

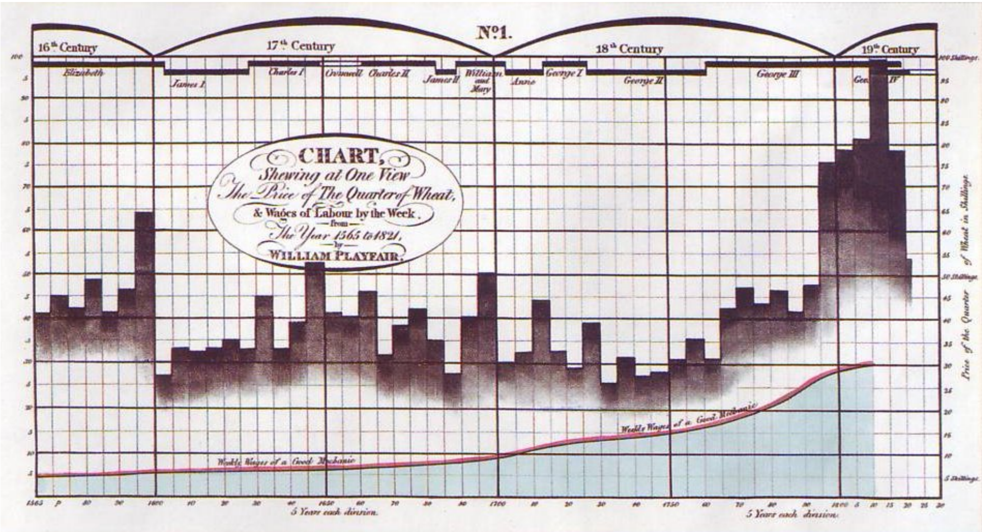



Figure 1.1: Price of wheat and weekly wages, by Playfair (1821).

*The effectiveness of Information Visualization hinges on two things:*
*it's ability to clearly and accurately represent information*
*and our ability to interact with it to figure out what the information means.*

— Stephen Few [Few, 2009]

## 1.1 CONTEXT

Computing technology has brought changes in the amount of information we capture and store in many areas of our lives, but also in the way we can represent it, process it visually, and interact with it. This has led to the emergence of a research field: Information Visualization.

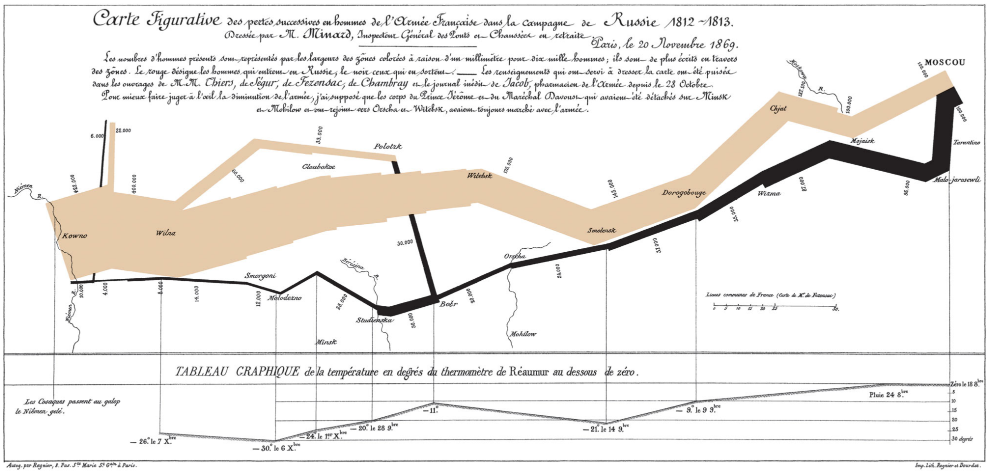



Figure 1.2: Minard's chart (1869) showing the number of human loss in Napoleon's 1812 Russian campaign, as well as their movements according to geographical landmarks and encountered temperatures.

### 1.1.1 *Information Visualization*

Information Visualization (Infovis) is *"the use of computer-supported, interactive, visual representation of (abstract) data to amplify cognition"* [Card et al., 1999]. This definition emphasizes four aspects of Infovis: visual representation, cognition, computer-support, and interaction.

*Visual representations* of information have been extensively used since the late 1700's [Tufte, 1986], after the invention of linecharts and barcharts by Playfair [Playfair, 1786] (e. g., Figure 1.1). The best illustration of this golden age of graphical representation is Napoleon's Russian campaign (Figure 1.2), considered as a graphical excellence and described as "the best graphic ever produced" by Tufte [Tufte, 1986].

Visual perception and *cognition* are interleaved: the human brain is particularly effective at processing visual information. Visual representations can be more effective than symbolic information because they amplify the cognition by providing the human more information, faster, and with less cognitive effort. Indeed, several visual features called preattentive are almost instantaneous to process by the human brain as opposed to symbolic information such as text [Treisman, 1985] requiring cognitive efforts. However, as illustrated Figure 1.3, if using a unique preattentive feature allows for high-speed visual processing, preattentive features can be in conflict. To optimize the (limited) use of preattentive features, the foundations of Infovis are based on studies of visual variables. Bertin organized the visual and perceptual elements of graphics according to the features and relations in data in his monumental *Sémiologie Graphique* [Bertin, 1967] before several researchers continue his work on efficient visual encodings for human perception [Card et al., 1999; Cleveland and McGill, 1984; Tufte, 1986; Ware, 2004].

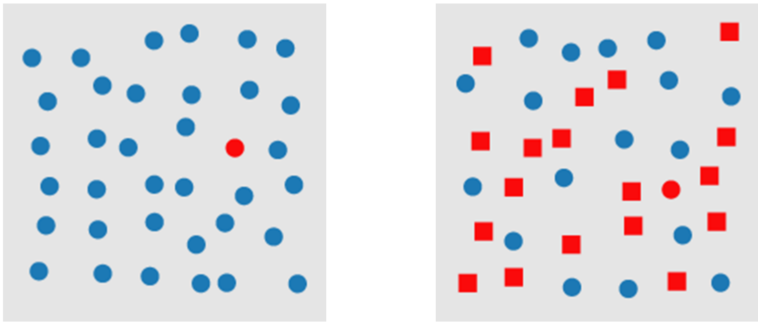


Figure 1.3: Finding the red circle is preattentive on the left because only one preattentive feature is used (hue); Finding it is not preattentive on the right because two preattentive features are used at the same time (hue and shape).

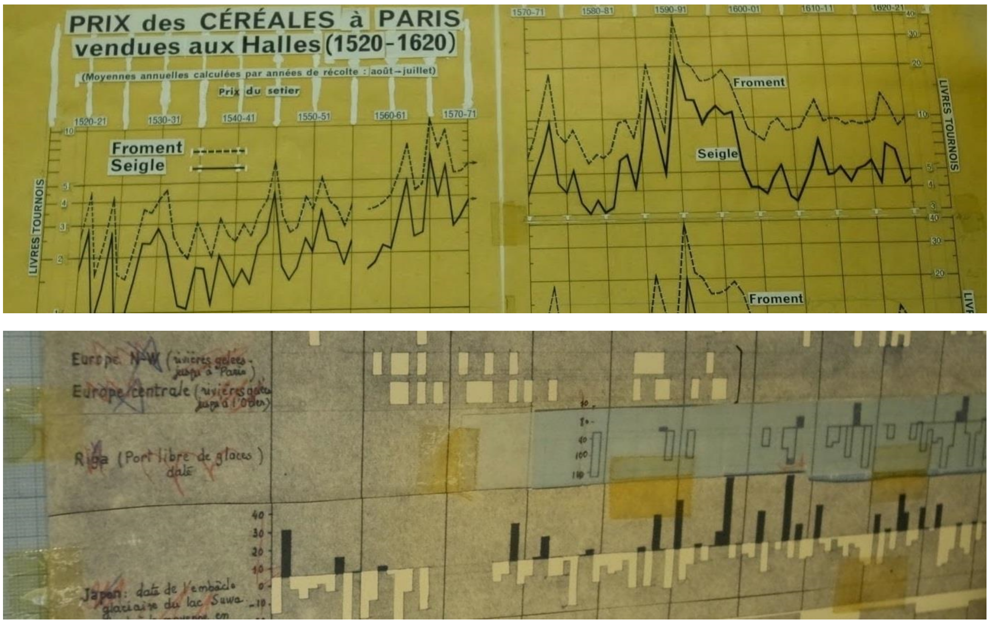



Figure 1.4: Examples of hand-crafted visual representations from the French National Archives, Baulant (top), Baehrel (bottom).

Visual representations moved to *computer displays* in the 1950's, as the technology offered new ways of constructing graphics by computer programs. While crafting visual representations on paper was a laborious manual process until then (e. g., Figure 1.4), computers provided an easy way of processing data and producing high-resolution graphics. Card et al. contributed to formalizing the field of Infovis [Card et al., 1999] and described the Infovis pipeline (Figure 1.5) as the reference model supported by computers to turn raw data into information, leading to insights: i) raw data is transformed into structured data; ii) data are mapped to visual marks or symbols with efficient visual encodings for human perception; iii) the visual marks are rendered on a screen, creating a view on the data; and iv) the user perceives the rendered image to perform his tasks.

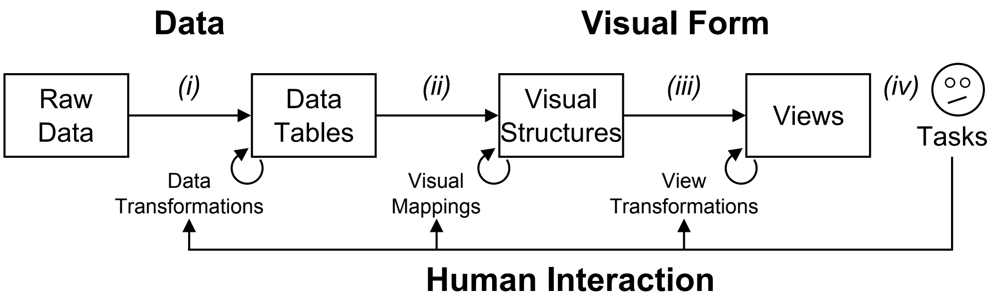



Figure 1.5: Card et al.'s Infovis pipeline [Card et al., 1999]: reference model for visualization.

If the visual representation, cognition, and computer-supported keywords of the definition of Infovis are thoroughly studied, the *interactive* component of the definition has received less emphasis from the visualization community [Yi et al., 2007]. However, interaction is the spine of the Infovis pipeline, each transition in the pipeline involving user interaction. The adage *"A picture is worth a thousand words"* is correct if and only if great attention has been paid at designing the picture, i. e. if the user applied meaningful successive transformations resulting in a valuable view on the data [Yi et al., 2007]. Bad pictures lead to unreadable information, or even worth to misleading interpretations that can involve wrong decisions [Fekete et al., 2008]. Interaction with the underlying data is a necessary step towards obtaining a good picture.

The importance of interaction for Infovis is widely recognized [Card et al., 1999; Chi and Riedl, 1998; Heer and Shneiderman, 2012; Jansen and Dragicevic, 2013; Lee et al., 2012; Pike et al., 2009; Thomas and Cook, 2005] but existing taxonomies for characterizing interactions for Infovis usually focus on describing at which stage of the Infovis pipeline the technique should occur [Chi, 2000; Chi and Riedl, 1998]. As advocated by Jansen and Dragicevic [Jansen and Dragicevic, 2013], taxonomies for interaction techniques try to answer one or several of the three following questions:

1. *What* is the user doing? (what is the effect of the interaction).
2. *Why* is she doing it? (what is the goal behind producing this effect).
3. *How* does she do it? (what are the means by which this effect is achieved).

The *Why* is dependant of Infovis tasks and has been largely addressed in task taxonomies [Amar et al., 2005; Dix and Ellis, 1998; Heer and Shneiderman, 2012; Kosara et al., 2003; Liu and Stasko, 2010; Shneiderman, 1996; Wegner, 1997; Wehrend and Lewis, 1990; Yi et al., 2007]. But the *What* and *How*—related to the action-perception loop between the user and the computer—have received less attention from the visualization community [Jansen and Dragicevic, 2013] (note that the *What* and *How* here do not have the same meaning than in [Brehmer and Munzner, 2013]). This thesis is at the intersection of both Infovis and Human Computer Interaction (HCI) to be a step towards this direction—with the perspective of improving visual representation techniques using new interactions.

### 1.1.2 *Advances in HCI*

In parallel to Infovis, the broader field of HCI has tremendously advanced since the age of conversational interfaces in characterizing *How* the user interacts with computers. Since the first Windows, Icons, Menus, Pointer (WIMP) interfaces, based on the Xerox Star interaction model [Smith et al., 1982], many researches have been devoted at reducing the gap between *Why* the user is doing something (intent) and *How* she accomplishes her tasks (execution of the intent) [Nielsen, 1993; Van Dam, 1997].

Modern interfaces are rooted in the direct manipulation model [Shneiderman, 1983], aimed at providing rapid, incremental and reversible actions on continuous objects of interest in order to facilitate learning and reduce the feeling of distance between the user and the computer [Hutchins et al., 1985]. Direct manipulation has then been applied to interfaces called post-WIMP interfaces, including instrumental [Beaudouin-Lafon, 2000], proxemics [Greenberg et al., 2011], touch-based [Wobbrock et al., 2009], and sketch-based interaction [Olsen et al., 2009], driven by the newly available technologies such as interactive tables, smartphones, wall-sized displays and tangible devices.

The field of HCI tends to focus on the person's perspective—interaction devices and techniques [Van Dam, 1997]: *Why* and *How* the user performs actions on the system. On the other hand, the Infovis community focuses on the user's intent (*Why*) and which feedback should provide the system, in terms of visual representation and view updates (*What*). From an Infovis perspective, there is an urgent need of considering the *Why*, the *What* and the *How* when designing interactive techniques in order to go further than the deprecated widget-based WIMP interfaces and fully take advantage of advances in HCI.

## 1.2 MOTIVATION AND APPROACH

Despite the newly available technologies, the desktop computer is still the ubiquitous device for visualizing data. While some researchers try to bring visualization beyond the traditional mouse and keyboard devices [Jansen and Dragicevic, 2013; Lee et al., 2012], desktop-based visualization shall survive if desktop interactions are made more intuitive, engaging and efficient.

### 1.2.1 *Desktop visualization is not dead*

Today, most Infovis systems remain WIMP interfaces and do not take advantage of these advances in HCI although researchers stressed the importance of interaction design [Lee et al., 2012] and fluid interaction [Elmqvist et al., 2011] in Infovis. Models such as instrumental interaction [Beaudouin-Lafon, 1997, 2000; Beaudouin-Lafon and Mackay, 2000] open the gates to designing new interfaces for new devices. But these models also give ways of empowering traditional devices such as the *mouse and desktop* computer. WIMP interfaces are known to be easy to learn because of their external consistency [Grudin, 1989], i. e. once the user learnt one WIMP interface, she can easily transfer her knowledge to similar interfaces. However, WIMP interfaces quickly become complex and difficult to use and Grudin advocates that ease of learn and ease of use can be in conflict [Grudin, 1989].

To highlight the importance of interaction in Infovis, it is necessary to have an overview of the different aspects of a visualization, from its creation to the generation of insights. van Wijk proposed an economic model to measure the value of a visualization technique [van Wijk, 2005], and he identifies the four following costs of a visualization:

- C1 *initial development costs* resulting from the development, implementation, and hardware acquisition needed to present the visualization.
- C2 *initial costs per user* capturing users' time investment to select and learn the visualization method and tailor it to their needs.
- C3 *initial costs per session* capturing the time and skills required to load a new dataset using the visualization.
- C4 *perception and exploration costs* capturing the users' time spent to perceive and understand the visualization, then to manipulate and explore the visualization to explore the underlying data.

C1 is clearly high for visualizations based on new hardware or technologies. This is less true for desktop visualizations, especially since visualizations are more and more online tools built on open-source libraries (e. g., d3.js [Bostock et al., 2011]).

C2 is related to learning. How the user discovers functionalities and how the user gets used to an interface is a whole body of research in HCI and related to factors such as interface consistency [Grudin, 1989] and affordances [Gibson, 1979; Norman, 1988]. Moreover, because ease of learn and ease of use may interfere, this thesis does not address this cost, based on the assumption that if the perceived benefit of the technique is high enough, then the user will take the time to learn it. As a simple example, users learn the gestures to scroll up and down using a touch-enabled phone while there is no affordance for the interaction technique.

C3 is almost unavoidable. Software use wizards to make this task easier, but research in this area is marginal [Baudel, 2006] yet clearly an important aspect of Infovis [Munzner, 2009; van Wijk, 2005]. Again, I assume that the data is well formatted and this thesis does not tackle this issue.

This thesis focuses on C4. Assuming that a visualization runs on a standard desktop computer with one mouse and one keyboard, that the dataset is well formatted, and that the user learnt the available interaction techniques. The remaining cost is the one of the data exploration leading to insights, by modifying and tuning the visualization parameters.

### 1.2.2 *Improving the Efficiency of Visual Representations*

In this dissertation, I address the general research question:

▷ How to improve the efficiency of existing desktop visual representations through effective interactions?

Infovis researchers devote huge efforts in designing new desktop visualization techniques, but only a few survive the test of time [van Wijk, 2005]. Instead of designing more new representation techniques, this work focuses on making existing ones more efficient through effective post-WIMP interactions.

I use the definition of post-WIMP interfaces—or post-WIMP interactions—from Van Dam who defines them as interfaces "containing at least one interaction technique not dependent on classical 2D widgets such as menus and icons" [Van Dam, 1997]. I apply this principle to make visualization techniques more efficient, while referring to direct manipulation principles and the descriptive and generative power of instrumental interaction. I describe several published projects exploring how post-WIMP interactions can improve visual representations in terms such as efficiency, ease of use, appropriation and playfulness, while maintaining user's state of flow [Bederson, 2004; Csikszentmihalyi, 1991].

## 1.3 THESIS STATEMENT

There is a tremendous effort from the Information Visualization community to design novel, more efficient or more specialized desktop visualization techniques. While visual representations and interactions are combined to create these visualizations, less effort is invested in the design of new interaction techniques for Infovis. In this thesis, I focus on interaction for Infovis and explore how to improve existing visualization techniques through efficient interactions. To become more efficient, the interaction techniques should reach beyond the standard widgets and WIMP user interfaces.

In this thesis, I argue that the design of novel interactions for visualization should be based on the direct manipulation paradigm and the instrumental interaction framework, and take inspiration from advanced interactions investigated in HCI research but not well exploited yet in Infovis. This philosophy is unusual as the standard approach consists of implementing standard widget-based interactions for new visual representation techniques, ensuring a short learning phase for the users but at the cost of tedious manipulations. I take the opposite approach, based on the assumption that if the perceived benefit of the interaction technique is high enough, then the user will take the time to discover it and learn it.

I explore design implications raised by novel interaction techniques, such as the tradeoff between cognitive congruence (the natural mapping between user's intent and action) and versatility (of the interaction techniques), the problem of engaging interaction (how to make the user engaged and willing to explore the visualization), and the benefits of seamless, fluid interaction. Finally, I provide guidelines and perspectives, addressing the grand challenge of building or consolidating the theory of interaction for Infovis.

## 1.4 THESIS OUTLINE

Chapter 2 (on page 11). In this chapter, I present the history of direct manipulation and extract a list of principles, benefits and challenges of direct manipulation interfaces. I also introduce some less explored aspects of interaction design. Then, I describe the role of interaction in Infovis and I present several examples of visualization techniques making use of direct manipulation, that I analyze using the previously extracted principles, benefits and challenges of direct manipulation interfaces. Finally, I present challenges and tradeoffs inherent to direct manipulation interactions for Infovis.

Chapter 3 (on page 39). In this chapter, I investigate how to improve the efficiency of an existing representation technique for multiple time series through effective interactions, and what are the benefits of such interactions. I detail the interaction techniques design, and I report empirical results from a user study showing that the proposed interactive visualization technique outperforms two static visual representations when the number of time series increases. I conclude by investigating the versatility of the proposed interactions techniques and propose implications for designing interactions for Infovis.

Chapter 4 (on page 65). In this chapter, I investigate the design of new interaction techniques for a standard visualization techniques: multivariate tables that evolve over time. I also explore how to integrate another visualization technique (a simple line chart) as an alternative view embedded into the table and accessible through interaction. Results from a crowdsourced user study show that the proposed interaction techniques are more efficient than standard time-navigation techniques for a large number of important tasks. Finally, I discuss design implications issuing from this project, and discuss among others user engagement and interaction techniques discoverability.

Chapter 5 (on page 89). In this chapter, I investigate what makes a time-navigation interaction technique direct. Based on the interaction techniques designed in Chapter 4, I present a novel interaction technique to navigate in time-dependant data graphics using transient trajectories of moving objects. I explore the design space of trajectory-based interaction and illustrate it with several case studies, and I propose a generic interaction model of the technique, applied to several standard visual representations. I conclude by providing a set of design guidelines for trajectory-based interaction techniques and discussing the benefits of improving the directness of time-navigation interactions.

Chapter 6 (on page 113). In this chapter, I present new coherent interactions for crafting tabular visualizations, based on Jacque Bertin's method for analyzing tabular data. Not only this chapter resurrects an ancient and forgotten visualization method, but it also emphasizes unexplored and important aspects of interaction for Infovis related to human-steered algorithms and the benefits of slow interaction. Results from a qualitative user study suggest that the proposed interactions make it possible for both scientists and a wider audience to explore, analyze and interpret their data, as well as to communicate their findings by visual means.

Chapter 7 (on page 149). In this chapter, I first summarize the work of this thesis. I present general guidelines, tradeoffs and perspectives that are emphasized by the different projects that were described in previous chapters. After summarizing the findings of this thesis, I lay out perspectives for future research. Specifically, I discuss the grand challenge of interaction techniques discoverability, and the need for an interaction mapping standard to improve discoverability, learning, and ease of use of interactive systems as both perspectives are factors of user engagement and visualization democratization.

# 2 BACKGROUND

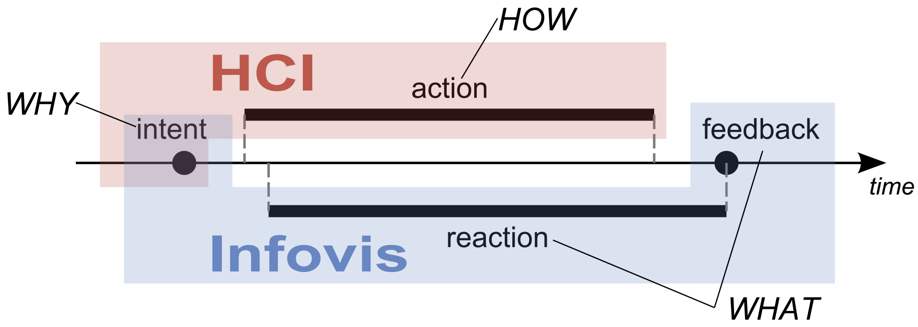



Figure 2.1: Definition of an interaction according to time. Adapted from [Lee et al., 2012] to illustrate which part of the interaction temporal spectrum HCI and Infovis focus on.

Lee et al. provide a simple definition of an interaction according to time [Lee et al., 2012] (see Figure 2.1): i) the person has an intention; ii) the person performs an action; iii) the system reacts to this action; iv) the system provides feedback to the person. As explained in Section 1.1 the field of HCI tends to focus on the person's perspective [Van Dam, 1997]: *Why* and *How* the user performs actions on the system. On the other hand, the Infovis community usually focuses on the user's intent (*Why*) and *What* feedback should provide the system in terms of visual representation and view updates.

Because this dissertation is at the intersection of both fields to encompass the whole spectrum of interaction, this chapter is organized as follows: I initiate the literature review with Section 2.1 presenting a history of direct manipulation, and highlighting principles, benefits and challenges that have been raised by the HCI community; and some broader challenges about less explored aspects of interaction. Then, Section 2.2 describes the role of interaction in the field of Information Visualization and presents several examples of visualization techniques making use of direct manipulation. Finally, Section 2.3 summarizes the background section and presents challenges and tradeoffs inherent to direct manipulation interactions for Infovis.

## 2.1 DIRECT MANIPULATION

This section provides an overview of the history of direct manipulation, since its introduction in the early 80th [Shneiderman, 1983, 1982]. Many researchers have discussed various aspects of direct manipulation over the past decades, leading to hot debates and controversies. The term 'direct manipulation' itself is difficult to interpret and the purpose of this section is to provide a background to the non-expert reader. After introducing the context of direct manipulation interfaces, this section organizes the work on the topic into principles, benefits, and challenges of direct manipulation.

### 2.1.1 *WIMP Interfaces*

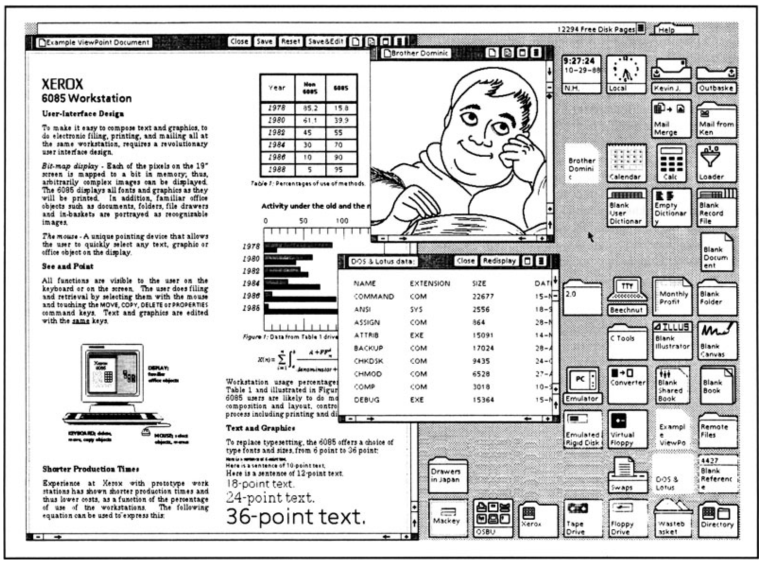



Figure 2.2: Xerox Star, the first WIMP interface [Smith et al., 1982].

Most existing modern user interfaces are "point-and-click" WIMP interfaces, which did not evolve much since the Xerox Star interface [Smith et al., 1982] (Figure 2.2) from the beginning of the 80th, made popular by the Macintosh. Widgets (e. g., buttons, menus or scrollbars) are the basic interactive tools of WIMP interfaces. What Van Dam noted in 1997 is still valid almost 20 years later: *"Applications today have much the same look and feel as the early desktop applications (except for the increased "realism" achieved through the use of drop shadows for buttons and other user interface widgets)"* [Van Dam, 1997].

The term *direct manipulation* was introduced in the early 80th [Shneiderman, 1983, 1982] when the interaction between a human and the computer started to move from conversational (e. g., command-line interfaces) to WIMP interfaces, and is still one of the basic paradigm of interface design today [Shneiderman and Plaisant, 2004]. Direct manipulation aims at making interaction more natural, intuitive, and predictable, resulting in easier to learn and use applications. Back then, direct manipulation changed the entire paradigm for HCI from *dialogue* to *manipulation* by introducing the illusion to manipulate an interactive world. Direct manipulation has now a long history that we summarize here in principles, benefits, challenges (or pitfalls).

### 2.1.2 *Principles*

Direct manipulation is characterized by four main principles [Shneiderman, 1983; Shneiderman and Plaisant, 2004]:

P1 Continuous representation of objects of interest.
P2 Physical actions (movement and selection by mouse, joystick, touch screen, etc.) or labelled button presses instead of complex syntax.
P3 Rapid, incremental, and reversible actions whose impact on the object of interest is immediately visible.
P4 Layered or spiral approach to learning that permits usage with minimal knowledge.

In another seminal article laying the foundations of direct manipulation, Hutchins et al. analyze Shneiderman's claims. They agree that direct manipulation interfaces can be more attractive and usable than conversational command-based interfaces but investigate deeper *what makes manipulation direct* [Hutchins et al., 1985] on a psychological basis. They propose to measure the degree of *directness* of an interface according to two factors: *distance*, and *engagement*.

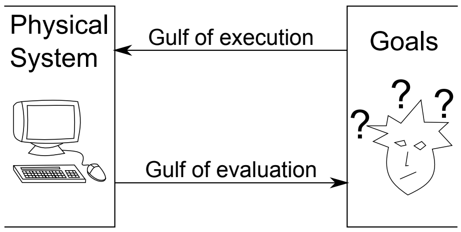



Figure 2.3: The two unidirectional gulfs of execution and evaluation, from [Hutchins et al., 1985].

*Distance* refers to the distance between a user thoughts about a problem (user task) and the way of achieving it using the computer (user interaction), i. e. the mental effort to translate goals into actions and evaluate their effects [Norman, 1986]. They introduce two types of *distance* between the user and the computer. The *semantic distance* refers to the information processing distance between the user's intentions and the facilities provided by the machine. The semantic distance is twofold, as illustrated Figure 2.3. It plays a role both in the gulf of execution—match between the interface language and the user's thought of the task, and in the gulf of evaluation—the amount of processing structure that is required for the user to determine whether the goal has been achieved. The *articulatory distance* concerns the relation between the input and output vocabularies of the interface language. To reduce articulatory distance, representations of objects provided by the system must behave as if they are objects themselves.

P5 Semantic directness in the gulf of execution.
P6 Semantic directness in the gulf of evaluation.
P7 Articulatory directness in the gulf of execution.

*Engagement* refers to the feeling of direct engagement, of first-personness, with the interface [Laurel, 1986], as opposed to conversational interfaces involving the user to stand in a third-person relationship to the objects of interest. Engagement is crucial for creating the illusion to manipulate the object, not the underlying program. Direct manipulation interfaces increase engagement when they ensure:

P8 Natural mapping to the task.
P9 Responsiveness, immediate feedback with no delays between execution and the results.
P10 No interference or intrusion.

In the following, I introduce more recent and less explored principles, derived from direct manipulation.

#### 2.1.2.1 *Instrumental Interaction*

Direct manipulation has then been applied to interfaces called post-WIMP interfaces, including instrumental [Beaudouin-Lafon, 2000], proxemics [Greenberg et al., 2011], touch-based [Wobbrock et al., 2009], and sketch-based interaction [Olsen et al., 2009], driven by the newly available technologies such as interactive tables, smartphones, wall-sized displays and tangible devices. Post-WIMP user interfaces also issue from direct manipulation but do not rely on menus, forms and toolbars but for example on gesture and speech recognition [Van Dam, 1997].

Beaudouin-Lafon introduces *instrumental interaction* in a series of articles [Beaudouin-Lafon, 1997, 2000; Beaudouin-Lafon and Lassen, 2000; Beaudouin-Lafon and Mackay, 2000; Beaudouin-Lafon et al., 2001] as a new interaction model, based on an analysis highlighting that WIMP interfaces do not follow the principles of direct manipulation. Indeed, WIMP interfaces introduce interface elements (widgets) that act as mediators between users and the object of interest [Beaudouin-Lafon, 2000]. Thus, this reduces the sense of engagement from the users because they manipulate directly intermediate objects, not the objects themselves. Instrumental interaction extends and generalizes the principles of direct manipulation by separating out the concept of an instrument as the mediator between users and domain objects, following the Activity Theory framework [Bødker, 1989]. The model is based on three properties to evaluate and generate instruments [Beaudouin-Lafon, 2000]:

The *degree of indirection* is a two-dimensional measure of the spatial and temporal distance introduced by instruments. the spatial distance refers to the physical distance between the instrument and the object of interest. The temporal distance measures the temporal interval between the action on the instrument and the information feedback. High temporal distances confuse users because they lose the causality between their actions and the system's responses [Michotte, 1963].

The *degree of integration* is the ratio between the degrees of freedom of the instrument and the input device. There is evidence that performance is impacted by the interrelationship between the perceptual structure of the task and the control properties of the device [Jacob et al., 1994]—what Grammel et al. call 'conceptual distance' [Grammel et al., 2013].

The *degree of compatibility* is a measure of similarity between actions on the instrument and the feedback on the object.

Beaudouin-Lafon argues that a good instrument should have a:

P11 Low degree of spatial indirection.
P12 Low degree of temporal indirection.
P13 High degree of integration.
P14 High degree of compatibility.

Although the previously enumerated challenges draw explicitly from the literature, several areas of interaction remain largely unexplored. In this section, I provide a small list of recommendations that capture these main unexplored—although crucial—areas of research both for interaction in general and for interaction for Infovis.

#### 2.1.2.2 *Flow and Seamless Interaction*

Flow is *"the condition in which people are so involved in an activity that nothing else seems to matter; the experience itself is so enjoyable that people will do it even at great cost, for the sheer sake of doing it"* [Csikszentmihalyi, 1991]. Bederson [Bederson, 2004] applies the theory of flow to interface design and differentiates novice and expert users in terms of skills and expected challenge. Green et al. argue that interaction is responsible for engaging the user and keeping the user in a continuous, uninterrupted 'cognitive flow' [Green et al., 2008]. Pike et al. argue that interaction should not be an afterthought when designing an interface, but the initial consideration. They emphasize that new interaction research is needed such that *"discovery is both natural and supported seamlessly by computational aids"* [Pike et al., 2009]. Elmqvist et al. also advocate for more careful interaction design and propose the design of 'fluid interaction' that promotes users to stay in the flow of their analytic process [Elmqvist et al., 2011]. It leads to the conclusion that 'good' direct manipulation should mean fluid interaction. Most of the design guidelines for fluidity they provide are reformulations of previously advocated direct manipulation principles: immediate feedback (P6, P9, P12), low indirection (P12, P11), and no interaction 'dead-end' (P3, P10). They also add the following principles:

P15 Smooth transitions between states to maintain user's flow and perceived continuity.
P16 Invisible interface to the user by integrating user interface components in the visual representation.
P17 Rewarding interaction (e. g., with animations, sounds and pretty graphics) to improve the user experience.
P18 No explicit mode changes by avoiding drastic visual changes and drastic interaction modality changes.

#### 2.1.2.3 *Congruent Interaction*

Hutchins et al. argue that semantic directness in the gulf of execution (P5) requests matching the level of description required by the interface language to the level at which the person thinks of the task [Hutchins et al., 1985]. Similarly, Frohlich notes that using commands which correspond closely to what the user want instead of model world actions may support easy and effective interactions [Frohlich, 1997], and Hutchins et al. add that reducing the semantic and articulatory distances provides a natural mapping to the task [Hutchins et al., 1985] (P8). I call this natural mapping between task and action *congruent interaction*, based on the congruence principle for graphics from Tversky et al.: *"The structure and content of the external representation should correspond to the desired structure and content of the internal representation."* [Tversky et al., 2002]. For example, since routes are conceived of as a series of turns, an effective external visual representation of routes will be based on turns; increases will be going upwards or to left or right; and increasing rates will be mapped to increasing slopes [Gattis, 2001; Gattis and Holyoak, 1996]. Actions that are congruent with thinking were also found to facilitate thinking [Segal et al., 2014].

This principle of congruence has already been applied to the fields of HCI and Infovis. Boy et al. define the level of *syntactic* congruence for the coupling of visual representation–task by the *"replaceability of the data-related terms by perceptual terms"* [Boy et al., 2014]. For example, a congruent question to the task "find the highest value in a bar chart" would be "find the highest bar"; conversely, the question "is the intersection between column A and row B highlighted?" is not congruent to the task "determine if A and B are connected in a matrix diagram" [Boy et al., 2014]. Other researchers studied the *cognitive* congruence. For animations, congruence means that there should be a natural correspondence between change over time and the conceptual information to be conveyed [Tversky et al., 2002]. For gestures, Segal found that congruent gestures for gestural interfaces improve cognition, performances and learning [Segal, 2011].

In terms of interaction design, I call *congruent interaction* an interaction whose action, reaction and feedback (*What* and *How*) are cognitively congruent to the intent of the user (*Why*). For example, temporal navigation in a video using a slider is not congruent to the task "find when this object in the scene is in the top-left corner of the screen". A congruent interaction is to grab the object on the scene and move it to the top-left corner using direct manipulation and observe the time change, as proposed in DimP [Dragicevic et al., 2008].

P19 Cognitively congruent interaction to the tasks it was designed for.

The more the enumerated principles are respected, the more direct manipulation is 'direct'. If an interface is a direct manipulation interface, then it inherits a number of benefits that I summarize in the next section.

### 2.1.3 *Benefits*

Shneiderman enumerates a first list of benefits that interfaces respecting direct manipulation principles should provide [Shneiderman, 1983]:

B1 Learnability: Novices can learn basic functionality quickly, usually through a demonstration by a more experienced user. Analogy, used as literary metaphor, is also effective for communicating complex concepts to novices [Halasz and Moran, 1982].
B2 Mastery: Experts can work extremely rapidly to carry out a wide range of tasks, even defining new functions and features, i. e. mastery of the system.
B3 Memorability: Knowledgeable intermittent users can retain operational concepts. They gain confidence in their capacity to retain mastery over time.
B4 Fewer error messages: Error messages are rarely needed.
B5 Better feedback: Users can immediately see if their actions are furthering their goals, and if not, they can simply change the direction of their activity.
B6 Reduced anxiety: Users experience less anxiety because the system is comprehensible and because actions are so easily reversible.
B7 Increased control: Users gain confidence and mastery because they initiate an action, feel in control, and can predict system responses.
B8 Enjoyment: Enjoyment using the system and eagerness to show the system to novices; desire to explore more powerful aspects of the system.

Hutchins et al. add that reducing the semantic and articulatory distances and encouraging the feeling of engagement involve [Hutchins et al., 1985]:

B9 Sense of control: Giving the user a sense of control by providing the user with the illusion that she is directly engaged with the semantic objects, not with the computer.
B10 Reduced effort: Reducing the effort required of the user to accomplish goals.
B11 Reduced learning: Reducing the amount of learning required.

In his provocative *The Case Against User Interface Consistency*, Grudin highlights both benefits and pitfalls of interface consistency that he characterizes using three types of consistency [Grudin, 1989]. *Internal consistency*, which often receives the most conscious attention of designers, refers to the consistency of a design with itself; *external consistency* to the similarity with other interface designs familiar to a user; and *consistency with the world beyond computing* to the analogic or metaphoric correspondence of a design to features in the world beyond the computer domain. Each type of consistency has pitfalls that we detail in Section 2.1.4, but also benefits:

B12 Improved learning: Internal consistency of an interface improves initial learning [Carroll, 1982].
B13 Comfort: Internal consistency of an interface improves ease of use [Carroll, 1982].

B14 Perceived quality: Internal consistency of an interface improves perceived quality [Carroll, 1982].
B15 Transfer of training: External consistency helps transfer of training [Grudin, 1989].
B16 Recall: Consistency with the world beyond computing may be significant aids to recall and initial learning [Grudin, 1989].

### 2.1.4 *Challenges*

20 years ago, Frohlich begins his history of direct manipulation with: *"One of the most significant developments of the 1980's in human computer interaction was the emergence of 'direct manipulation' as a theoretical concept and design practice. [Direct manipulation] has had a massive and largely beneficial impact on the face of personal computing and continues to exert a strong influence on the design of interactive software today."* [Frohlich, 1993].

The fact is that, despite being largely criticized, direct manipulation interfaces still rule the world today and Frohlich's introduction is still mostly valid. Although the benefits of direct manipulation are accepted by the HCI and Infovis communities, several researchers have stressed the drawbacks of direct manipulation since its introduction. Here we provide an extensive review of the challenges raised and the controversies against direct manipulation pointing out that the paradigm is not a panacea.

Hutchins et al. question some of the original claims for direct manipulation interfaces:

C1 *Learnability* and *memorability* depend upon the user's prior knowledge and its semantic mapping to the system. Shneiderman also stressed the challenge of learnability, explaining that users must learn the meaning of the components of the graphic representation [Shneiderman, 1983].
C2 Expert performance slows down through direct manipulation because of the tedious manipulations it may involve.
C3 Ease of use is not directly related to the directness of the interface: difficulty can lie in the task domain, not in the interface domain.

Frohlich criticizes direct manipulation for its deficiencies in managing abstract or meta-level operations [Frohlich, 1993, 1997]. He enumerates 7 challenges for direct manipulation that were also underlined by other researchers:

C4 Referring to previous actions.
C5 Scheduling actions. Whittaker adds that scheduling such as reminders or automatic behavior such as e-mail auto-response are not compatible with the immediately visible impact of actions (P3) [Whittaker, 1990].
C6 Targeting unseen objects [Claasen et al., 1990; Hutchins et al., 1985]. As noted by Whittaker, locating objects that are not visible on the screen is challenging, and even impossible when the user does not know the location of the target because of the continuous representation of the objects of interest (P1) [Whittaker, 1990].

C7 Selecting groups of objects.
C8 Performing repetitive actions. Whittaker emphasizes that incremental operations (P3) are tedious because they prevent automatic application of repetitive actions to a set of objects [Whittaker, 1990]. Gentner and Nielsen adds that direct manipulation requires to operate at the atomic level, but that it is necessary to be able to trigger series of actions at once or apply an action on several objects of interest by a unique interaction [Gentner and Nielsen, 1996].
C9 Performing concurrent actions.
C10 Specifying precise actions or values. Pointing with great precision also have a high articulatory indirectness [Hutchins et al., 1985] (conflicting with P7).

Van Dam states that one of the advantages of WIMP Graphical User Interface (GUI)s is their ease of use, or how 'user-friendly' they are [Van Dam, 1997]. However, the author mixes ease of use and ease of learning: he calls ease of use for novices their ability to learn and remember the interface, and for power users their effort required to be highly productive. As explained later by Grudin, ease of learning and ease of use are two distinct factors and they must be considered separately as they can be conflicting [Grudin, 1989]. Actually, WIMP interfaces are only easy to learn because once a user masters a WIMP interface, she can easily transfer her knowledge to another WIMP interface thanks to their external consistency (B15) [Grudin, 1989]. Merging ease of use and ease of learning is a frequent mistake. For example, Zhao et al. write about direct manipulation: *"physically manipulating graphical representations of objects has the benefit of being easy to use and learn"* [Zhao et al., 2011].

C11 Ease of learning and ease of use are independent dimensions that can be conflicting.

The use of metaphors to facilitate learning is one of the basic principles of direct manipulation and is related to affordances. The concept of affordance was first coined by Gibson [Gibson, 1979] and introduced to the HCI community by Norman [Norman, 1988]. According to Norman, an affordance is the design aspect of an object which suggest how the object should be used; a visual cue to its function and use. Norman provides many illustrative examples where visuals greatly assist task performance and efficiency. He argues that it is vital to match the representation used in a visualization to the task it is addressing.

In the field of HCI, affordances are related to the use of real-world and physics metaphors. The use of metaphors in interfaces is advocated by many researchers to facilitate learning [Ahlberg and Shneiderman, 1994; Hill, 1995], guide users, transfer real-world knowledge [Faulkner, 1998; Hackos and Redish, 1998; Mandel, 1997], and exploit the capability of the human eye [Dix and Ellis, 1998]—for a valuable history of metaphors in HCI, see [Blackwell, 2006]. However, Halasz and Moran argue that literary metaphors are *"just a communication device meant to make a point in passing. Once the point is* *made, the metaphor can be discarded".* They add that for detail reasoning about computer systems, analogy should not be used but abstract conceptual models [Halasz and Moran, 1982].

While interface consistency can help users learn interfaces, it can also be a pitfall. In particular, Consistency with the world beyond computing (B16) can break down at some point because analogies are weaker correspondences [Grudin, 1989; Helander, 1988] and the computer provides opportunities to add more 'magical' functionality which does not rely on real-world metaphors [Smith, 1987]. Halasz and Moran add that teaching a system to novices is more efficient than just using metaphors to introduce them to a new system [Halasz and Moran, 1982]. About external consistency, Few also wrote *"People often develop strong preferences for visualizations that perform poorly, merely because they are familiar or superficially attractive"* [Few, 2011], moderating B15.

- C12 Metaphors can be discarded once the interaction has been learnt.
- C13 Metaphors are not the most efficient way of learning a system: novices should be introduced to the interface by teaching in order to ensure B1 (conflicting with B16).
- C14 Metaphors can be superficially attractive.

If each individual component (i. e. widget) of a WIMP interface is easy to use and learn, the more complex the application—the interface in its whole, the nonlinearly harder it is to use. In front of a complex WIMP interface, the user gets lost in the functionalities and will eventually use only a subset of the possibilities [Van Dam, 1997]. Moreover, the interface complexity forces the user to spend more time manipulating the interface than the application itself, i. e. struggling with the interface components instead of focusing on his or her task. Then, WIMP interfaces do not scale for more complex applications, adding more and more widgets taking too much real screen estate and confusing the user [Shneiderman and Maes, 1997]. This is unsurprising and related to the famous principle from Dieter Rams *"Good design is as little design as possible - Less, but better because it concentrates on the essential aspects, and the products are not burdened with non-essentials. Back to purity, back to simplicity"* [1].

- C15 The more the directly manipulable widgets, the harder the interface is to use.

Interaction is also difficult when the interaction space is multidimensional. 3D interaction is an important and recognized challenge (e. g., in Scientific Visualization). Considerable efforts are dedicated to this problem, such as 2D interaction metaphors in a 3D virtual environment using a mouse (e. g., [Angus and Sowizral, 1995; Balakrishnan et al., 1997]), multi-finger gestural interaction (e. g., [Grossman et al., 2004]), and 3D widgets and menus (e. g., [Conner et al., 1992; Dachselt and Hübner, 2007]). Jankowski and Hachet propose a recent survey of interaction techniques for 3D environments

1 https://www.vitsoe.com/gb/about/good-design

[Jankowski and Hachet, 2012]. Interaction with 3D environments is outside the scope of this dissertation. However, multidimensional does not mean 3D, and Infovis often deals with non-spatial multidimensional data, with larger numbers of dimensions than three. As noted by Lee et al., *"Infovis often deals with multidimensional data. The mapping of multidimensional data tasks to 2D widgets may not be particularly natural. An example where this becomes evident is widgets for 3D navigation in information spaces [Van Dam, 1997]"*.

C16 2D interaction are barely convenient for multidimensional information spaces.

Another important challenge is human-steered algorithms—making algorithms interactive by establishing a communication loop between the user and the algorithm. Tweedie argues that all algorithmic transformations provide indirect manipulation and most software provide algorithms as automatic tools acting as black boxes [Tweedie, 1997]. Eisenberg points out that *"We might find in practice, that an algorithm rich in interactive capabilities actually outperforms (in the quality of solutions) a more time efficient 'black box algorithm"'* [Eisenberg, 1996], and few systems allow steered interaction with algorithms (e. g., VisDB [Keim and Krigel, 1994] and Hypergami [Eisenberg, 1996]). Interactively steering algorithms is an established challenge in the Visual Analytics community; but less in Infovis. However, in Infovis, human-steered algorithms are needed for example for layout algorithms (e. g., graph layout, treemap layout, matrix reordering) and color selection algorithms. A dialogue is needed between the user and the system to iteratively steer the algorithm. For example, Tableau proposes visual representations algorithmically but user interactions are needed to explore alternatives and steer the result [Mackinlay, 1986; Tableau Software, 2014]

C17 Human-steered algorithms are an important and underexplored aspect of interaction for Infovis.

### 2.1.5 *Summary*

The principles, benefits and challenges derived from direct manipulation are summarized in Table 2.1. I use three intersecting groups: Learning, Ease of use, and Seamless and fluid interaction. These three groups are then organized in categories.

Metaphors and interface consistency impact *learning* in terms of initial learning, recall, retaining mastery and transfer of training. Congruent interactions provides a natural mapping to the task and impacts both *Learning* and *Ease of use*, the latest depending also on the efficiency of the interaction, i. e. fast and comfortable. Incremental actions and immediate feedback is at the intersection of *Ease of use* and *Seamless and fluid interaction* and allows for changing the user's activity direction. *Seamless and fluid interaction* also depends on: the feeling of control that impacts mastery of the system; and the feelings of playfulness and engagement that reduce anxiety and make the user eager to explore more powerful aspects of the system.

Table 2.1: Summary of direct manipulation principles, benefits and challenges according to *Learning*, *Ease of use*, and *Seamless and fluid interaction*, and organized in categories.

| | Category | Benefits | Principles | Challenges |
|---|---|---|---|---|
| LEARNING | Metaphors | •B1 Effective for novices<br>•B11 Effective for learning<br>•B16 Effective for recall | •P4 Using metaphors | •C12 Superficially attractive<br>•C13 Less efficient than teaching<br>•C14 Useful only at the beginning |
| LEARNING | Consistency | •B3 Confidence in retaining mastery<br>•B15 Transfer of training | •B12 Internal consistency improves learning<br>•B15 External consistency helps transfer of training | •C1 Learning depends on the user's prior knowledge |
| LEARNING / EASE OF USE | Congruent interaction | •P8 Provides a natural mapping to the tasks | •P2 Physical actions<br>•P5 High semantic directness<br>•P11 Low degree of spatial indirection<br>•P13 High degree of integration<br>•P14 High degree of compatibility<br>•P19 Interaction congruent to the tasks | •C16 2D interaction for multidimensional information spaces<br>•C11 Ease of learning and ease can be conflicting |
| EASE OF USE | Efficiency | •B2 Experts can work extremely rapidly<br>•B10 Reduces the effort required of the user to accomplish goals | •P7 High articulatory directness<br>•B13 Internal consistency<br>•C15 Few widgets | •C3 Difficulty can lie in the task domain, not the interface domain<br>•C7 Selecting groups of objects<br>•C10 Performing precise actions |
| EASE OF USE / SEAMLESS AND FLUID INTERACTION | Incremental actions, immediate feedback | •B5 Users can immediately see if their actions are furthering their goals, and change the direction of their activity | •P1 Continuous representation of the objects of interest<br>•P3 Rapid, incremental, reversible actions<br>•P6 Rapid visual evaluation<br>•P9 Immediate feedback<br>•P12 Low degree of temporal indirection | •C4 Referring to previous actions<br>•C5 Scheduling actions<br>•C6 Targeting unseen objects<br>•C8 Performing repetitive actions, applying an action on several objects by a unique interaction<br>•C9 Performing concurrent actions |
| SEAMLESS AND FLUID INTERACTION | Feeling of control | •B7 Users gain confidence and mastery because they feel in control | •P5 Semantic directness in the gulf of execution<br>•P15 Using smooth transitions between states<br>•P16 Making the interface invisible<br>•P18 Avoiding explicit mode changes<br>•B9 Illusion to manipulate the semantic objects | •C2 Expert performance slows down because of the tedious manipulations<br>•C17 Human-steered algorithms |
| SEAMLESS AND FLUID INTERACTION | Playfulness and engagement | •B6 Reduces anxiety<br>•B8 Makes the user eager to show the system to novices and explore its powerful aspects | •P10 Non interfering interface<br>•P17 Rewarding interaction<br>•B4 No error messages<br>•B14 High internal consistency | |

## 2.2 INTERACTION FOR INFORMATION VISUALIZATION

*While existing research in the area often focuses on representation, we highlight the overshadowed, but very important interaction component and strongly argue that it provides a way to overcome the limits of representation and augment a user's cognition.*

— Yi et al. [Yi et al., 2007]

Dix and Ellis explain that simple yet effective interactions greatly enhance static visual representations by increasing user's understanding without the need for extensive training or expensive implementation [Dix and Ellis, 1998]. They show how several standard visual representations are made more efficient by augmenting them with simple interactions.

In the following subsections, we refer to *What* the user is doing, *Why* she is doing it, and *How* she does it introduced in Section 1.1.1 to characterize interaction in Infovis.

### 2.2.1 *Interaction Classifications*

The *Why* has received the most attention from the Infovis community. Many taxonomies and interaction frameworks in visualization refer to the original Infovis pipeline [Card et al., 1999; Chi and Riedl, 1998] (Figure 1.5) and its refined versions [Carpendale, 1999; Jansen and Dragicevic, 2013; Spence, 2007; Tobiasz et al., 2009]. This is unsurprising as the primary function of interactivity is to allow users to alter the pipeline at different stages in their data exploration process [Card et al., 1999]. Thus, these taxonomies focus on the intent of the users (*Why*)—related to the tasks—and sometimes to the system reaction and feedback (*What*)—related to the dynamic visual representation.

Taxonomies usually decompose user intents into elementary analytical tasks [Amar et al., 2005; Kosara et al., 2003; Liu and Stasko, 2010; Shneiderman, 1996; Wehrend and Lewis, 1990; Yi et al., 2007]. For example, Card et al. describe which interaction can be applied at each stage of the pipeline [Card et al., 1999]. On the opposite, Chi and Riedl start from the Infovis pipeline to classify all possible operations according to the pipeline and indicate that the closer an operator is to the view in the pipeline, the higher the possible amount of direct manipulation [Chi and Riedl, 1998]. Both taxonomies describe *Why* each interaction has value and *What* should be the result of the interaction, without specifying *How*—which interaction technique implementation is needed.

Yi et al. make an extensive review of interaction techniques for Infovis [Yi et al., 2007] that they categorize based on the user's intent (*Why*) as an initial step towards defining a science of interaction to support visual analytics [Thomas and Cook, 2005]. They identify taxonomies of low-level interaction techniques [Buja et al., 1996; Chuah and Roth, 1996; Dix and Ellis, 1998; Keim, 2002; Shneiderman, 1996; Wilkinson, 2005], taxonomies of user tasks

[Amar et al., 2005; Zhou and Feiner, 1998], a taxonomy of interaction operations [Ward and Yang, 2004], and taxonomical dimensions of interaction techniques [Spence, 2007; Tweedie, 1997]. Heer and Shneiderman proposed a taxonomy of 12 task types grouped into three high-level categories [Heer and Shneiderman, 2012]: data & view specifications; view manipulation; and process & provenance. In both articles, the authors discuss the goals of the users (*Why*), and provide some examples of actions (*How*) that might be needed to affect the system's response (*What*).

To summarize, the *Why* has been extensively studied in Infovis, and the *What* to a lower extent. However, the *How* is usually not considered as a first thought in the community while user's actions can occur at any stage of the Infovis pipeline. The next subsection explores previous work at the intersection of HCI and Infovis.

### 2.2.2 *Direct Manipulation in Infovis*

There have been dramatic progresses in research beyond mouse and keyboard for Infovis, and several recent systems are based on new interaction modalities such as gestural and multi-touch interaction (e. g., [Baur et al., 2012; Frisch et al., 2009; Isenberg et al., 2010; Jetter et al., 2011; North et al., 2009; Schmidt et al., 2010; Vlaming et al., 2010; Voida et al., 2009]), sketch-based interaction (e. g., [Browne et al., 2011; Holz and Feiner, 2009; Ryall et al., 2005; Wattenberg, 2001]), speech recognition (e. g., [Cox et al., 2000; Sun et al., 2010]), and tangible and physical visualizations (e. g., [Cockburn and McKenzie, 2002; Jacob et al., 2002; Jansen, 2014; Jansen et al., 2013; McGookin et al., 2010]).

Conversely, most Infovis systems and interactions are still WIMP interfaces and buttons, menus and other standard widgets are the exclusive interaction modalities despite the fact that the field of HCI has tremendously progressed since then. Although WIMP interfaces are direct manipulation interfaces, their degree of directness along the directness continuum is low, and many of the identified principles and benefits are not respected.

Nonetheless, a few systems attempt to leverage these advances in HCI, proposing new on-desktop interaction techniques with respect to the direct manipulation principles and benefits extracted in Section 2.1. In this section I provide a selection of these systems to analyze how each principle can be applied, each benefit evaluated, each challenge highlighted or solved, and more importantly which tradeoffs of new problems appear.

#### 2.2.2.1 *Dynamic queries*

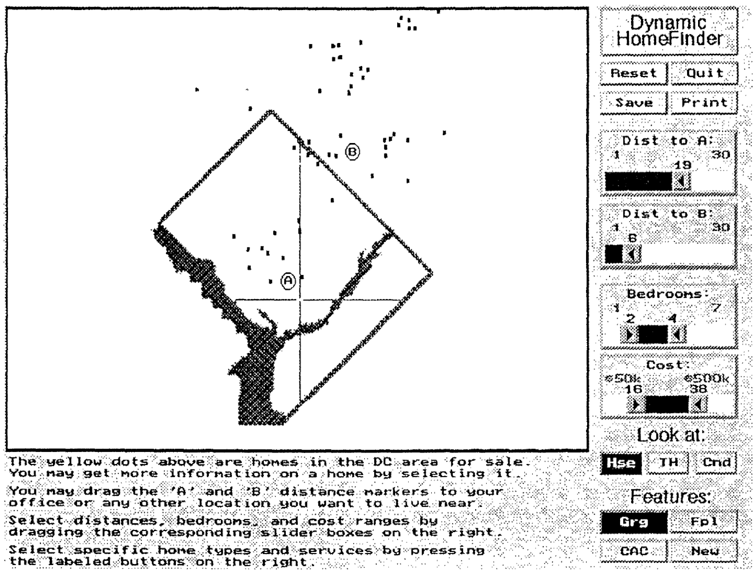



Figure 2.4: The Dynamic HomeFinder query system [Ahlberg et al., 1992]. Queries are described by sliders and buttons.

The *Dynamic Homefinder* [Ahlberg et al., 1992; Shneiderman, 1994; Williamson and Shneiderman, 1992]—and similarly, the FilmFinder [Ahlberg and Shneiderman, 1994]—is one of the first applications of direct manipulation to Infovis. The interface shown Figure 2.4 presents both a query panel (right) and a map (left) where dots represent individual homes. The queries are made by the user by moving sliders such as cost, number of bedrooms, and distance to locations of interest. The dots on the map are updated continuously when one of the widgets is manipulated.

This interface is a perfect illustration of principles P1, P3, P6, P9, and P12. It fulfills the *Incremental actions, immediate feedback* category, but does not face any of its challenges. The system is also easy to learn: it uses *metaphors* (P4) that are effective for learning (B11) and recall (B16), especially for novices (B1). The system has a high internal consistency (B12), and even though dynamic query widgets were new at this time, their extensive use today makes these interaction modalities externally consistent (B15); thus fulfills the *Consistency* category. However, the interactions are not congruent to the tasks, thus the mapping is not natural (P8). A slider only has one degree of freedom when the mouse has two, thus the degree of integration is 1/2 (P13), and the widgets have a high spatial offset (P11). Finally, the interactions are 1D when the query result is 2D, making the interaction not congruent to the tasks (P19).

To summarize, this system uses metaphors and internal consistency to improve initial learning, external consistency to facilitate transfer of training and mastery retaining, and incremental actions with immediate feedback to make the system both easy to use and seamless. However, the interactions are not congruent, while this category is at the intersection of learning and ease of use.

#### 2.2.2.2 *The reorderable matrix*

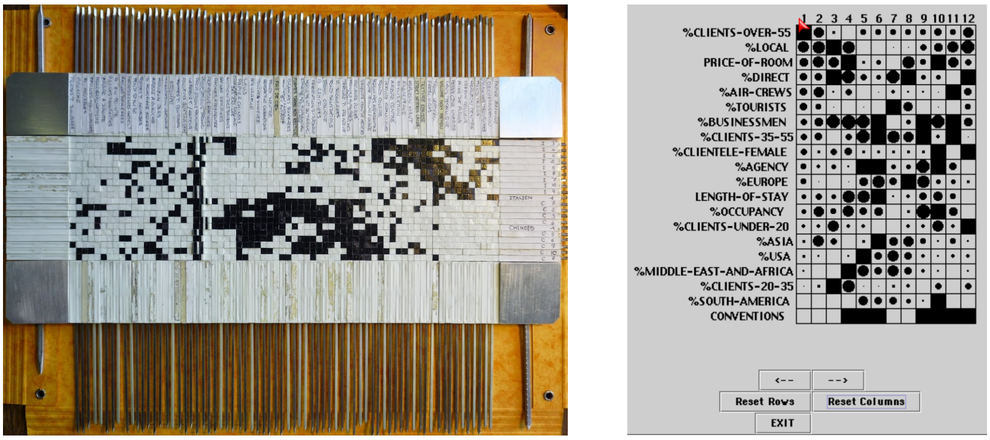



Figure 2.5: The reorderable matrix. On the left one of Bertin's physical matrices; on the right an implementation of the reorderable matrix by drag and drop of rows and columns [Siirtola, 1999].

Before the Infovis research field was born, and even before the computers were standardized, Bertin argued for direct manipulation of what he called 'mobile images' for data exploration, analysis and communication [Bertin, 1975]. More specifically, he spent years refining a method to reorder matrices manually by rearranging rows and columns (Figure 2.5, left). Several researchers attempted to resurrect the method by transposing it to computers, leveraging the advances in HCI and Infovis (e. g., [Bertin and Chauchat, 1994; Gimeno and coise Durand, 1988; Sawitzki, 1996; Siirtola, 1999]). An extensive review of systems implementing Bertin's method is provided Section 6.1. In particular, Siirtola proposes to rearrange rows and columns using drag and drop (Figure 2.5, right) to preserve the original physical metaphor [Siirtola, 1999] in order to facilitate learning (P4, B1, B11, B16). When rearranging rows and columns are the only available interactions, the interface is also internally consistent (B12) and users can retain mastery over time (B3). The interaction is also congruent to the task (P19): to discover groups of rows or columns, the user physically moves rows and columns (P2, P5, P11, P14). The method is easy to use because of *incremental actions, immediate feedback* (P1, P3, P6, P9, P12).

However, despite a high internal consistency (B13) and no widgets at all (C15), the *efficiency* of the method can be low: the experts do not work extremely rapidly (B2), and manipulating the matrix is tedious (B10)—according to Bertin, manipulating a matrix manually could take weeks because the difficulty lies in the task domain, not the interface domain (C3).

On one hand, it is a perfect example of *feeling of control*: the slow manipulation of data generates insights, a virtue of the method argued by Bertin [Bertin, 1975]. The interface is invisible to the user (P16) who directly manipulates the semantic objects (B5, B9) without any mode changes (P18). On the other hand, the method suffers from the tedious manipulations it involves (C2). Automatic reordering algorithms can limit the tedious manipulations, but it may reduce the amount of insights generated and may break the flow of the user. It emphasizes the need for interactions to steer algorithms (C17), in order to have the advantages of both the efficiency of algorithms to accelerate the process (C2) and manual manipulation to generate insights.

To summarize, the reorderable matrix is very easy to learn and recall, and easy to use. The manual interaction (physical or by drag and drop) to reorder rows and columns is congruent to the task. The main lesson from this example is that there is a tradeoff between *efficiency* and *feeling of control* that can be solved by human-steered algorithms (C17) to accelerate the process (C2) while generating insights. We tackle this challenge in Chapter 6.

#### 2.2.2.3 *Parallel coordinates*

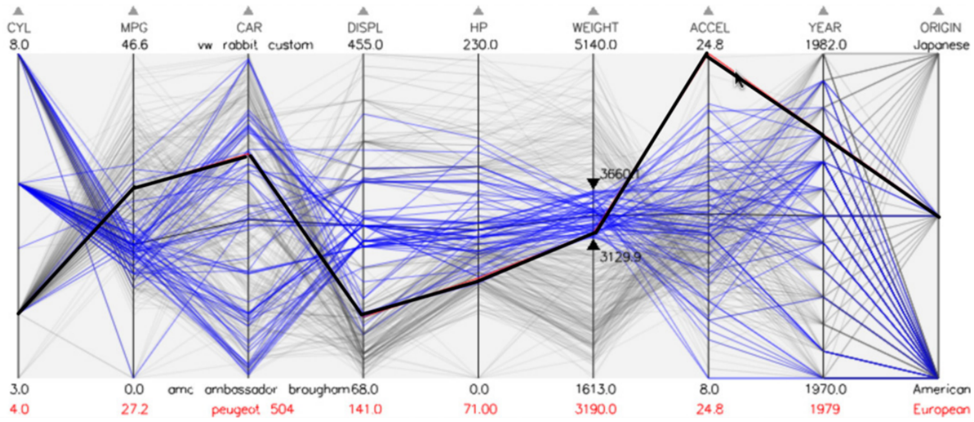



Figure 2.6: Brushing axes in parallel coordinates [Siirtola and Räihä, 2006].

Parallel coordinates are a two-dimensional technique to visualize multidimensional datasets [Inselberg, 1985; Inselberg and Dimsdale, 1990]. Parallel coordinates are one of the most popular visualization techniques where attributes are represented as axes and data items are represented as lines linking the axes. They have been thoroughly studied and Siirtola and Räihä provide a good review of interaction techniques for parallel coordinates [Siirtola and Räihä, 2006]. Most known techniques are brushing and rearrangement by direct manipulation.

Brushing is a physical action (P2) that can be performed by brushing one or several axes to filter data items according to these dimensions (Figure 2.6). Brushing is a two-dimensional interaction for filtering multi-dimensional data (C16): drawing a 'tunnel' for multi-dimensional brushing by selecting the polylines that fall completely within it [Martin and Ward, 1995]; and angular brushing to select polylines according to their slope between two axes [Hauser et al., 2002]. Although brushing is said to be a direct manipulation technique, these examples raise the problem of defining the object of interest. Multi-dimensional brushing by drawing a tunnel is direct if the object of interest is the data objects because the visual query is similar to the polylines representation, both spatial (P11) and temporal offsets (P12) are low, and the degree of integration (P13) and compatibility (P14) are high. Conversely, brushing the axis is direct if the object of interest is the axis itself, which is the case when one is focusing on a particular dimension. That is what Tweedie calls mechanized direct manipulation [Tweedie, 1997]. Finally, angular brushing is an abstract operation where the object of interest is not the data objects nor the axes, but the correlation between dimensions. Angular brushing is congruent to the associated task (P19, P5, P8, P14): to filter according to slope, the user draws an angle.

Rearrangement consists of reordering the axes of the visualization by drag and drop and change their direction to produce different views of the same dataset. Again, the manipulation is fully direct if the object of interest is the axis to be manipulated, not the polylines.

Kandogan proposes Star coordinates [Kandogan, 2001], inspired by both parallel coordinates and the reorderable matrix, for visualizing multidimensional datasets. Conversely to parallel coordinates, the axes are not parallel but laid out radially, and data points are positioned in the 2D space according to their value for each dimension. The technique proposes several interaction techniques by direct manipulation of the axes: brushing an axis to highlight data points whose values according to the associated dimension fall into a range, similarly as for parallel coordinates; rotating an axis to relayout the data points; and scaling an axis to change its influence on the result. All these operations are performed directly on the axes, not the data points.

Parallel coordinates and Star coordinates raise the challenge of identifying the object of interest and the associated interactions. They are also good examples of seamless interaction in terms of *feeling of control*: users gain mastery (B7) because of semantic directness in the gulf of execution (P5) and the illusion to manipulate the semantic objects (B9) because the interface is invisible (P16).

The interactions are not consistent and the interface difficult to learn, with no obvious affordances. However, the interactions are highly congruent to the tasks and the interfaces have a high *efficiency*, experts working extremely rapidly with the tools—although performing precise actions remains difficult (C10). This is a good illustration of C12, where users have to learn the interactions without affordances but are very efficient once these are mastered (C11).

#### 2.2.2.4 *Surrogate objects*

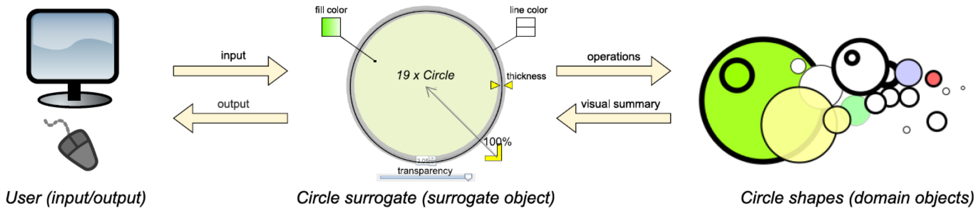



Figure 2.7: The surrogate object as a mediator between the user and the domain objects. From [Kwon et al., 2011].

Kwon et al. propose surrogate objects to address three of the previously identified challenges for direct manipulation (C10,C9, and C6) [Kwon et al., 2011]:

- Access (C10): manipulating small, distant, or attribute-rich objects under limited space, high density, or high precision.
- Multiple objects (C9): Manipulating multiple objects simultaneously as a group (including group attributes).
- Intangible properties (C6): Manipulating intangible object properties (abstract properties with no visual form).

Interaction is done through surrogate objects, which act as mediators between the user and the domain objects (Figure 2.7). Then, it achieves a high degree of compatibility (P14) by allowing users to manipulate more interface components using the surrogate object than the domain object itself (C6). It respects P1, P3, P6, P9, and P12. Moreover, a surrogate object can be the mediator for several distinct objects, allowing for group manipulation (C7, C8, C9). Because the physical actions (P2) on the mediator are identical to the ones performed on the object of interest directly, it also ensures P5 and P6. The tradeoff with surrogate objects is that they introduce spatial indirection (P11) between user and domain objects. This highlight a tradeoff between spatial indirection and compatibility in the interface for congruent interaction (P19).

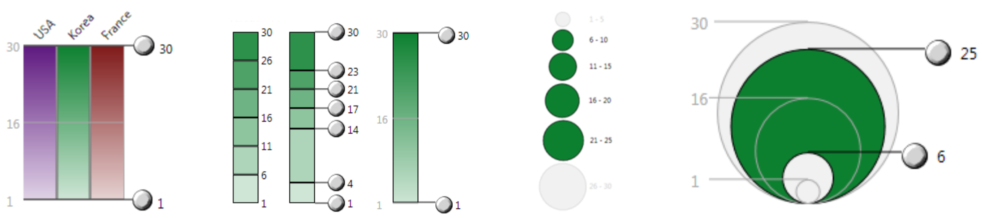



Figure 2.8: Interactive legends as a surrogate object for filtering in visualization (from [Henry Riche et al., 2010]).

Another example of surrogate interaction is the interactive legends, advocated by Tufte [Tufte, 1986] and applied to cartography [Peterson, 1999]. Interactive legends were found to be faster for perceiving data values by representing visual elements that can be filtered and selected directly in the visual representation [Henry Riche et al., 2010] (Figure 2.8). Along this line, several researchers (e. g., [Andrienko and Andrienko, 1999; Eick, 1994; Tweedie et al., 1994]) proposed to embed visual representations in dynamic query widgets, such as histograms showing the distribution of data for sliders. Thus, if the global object of interest is the result of the query, the augmented widget becomes the object of interest during the filtering process and the spatial offset is null (P11).

To summarize, surrogate objects introduce spatial indirectness (P11) by using a mediator between the user and the domain object but deal with the challenges of group manipulation and manipulation of non visible object properties. Introducing spatial directness allows for more efficient, congruent, and incremental interactions.

#### 2.2.2.5 *Dust & Magnet*

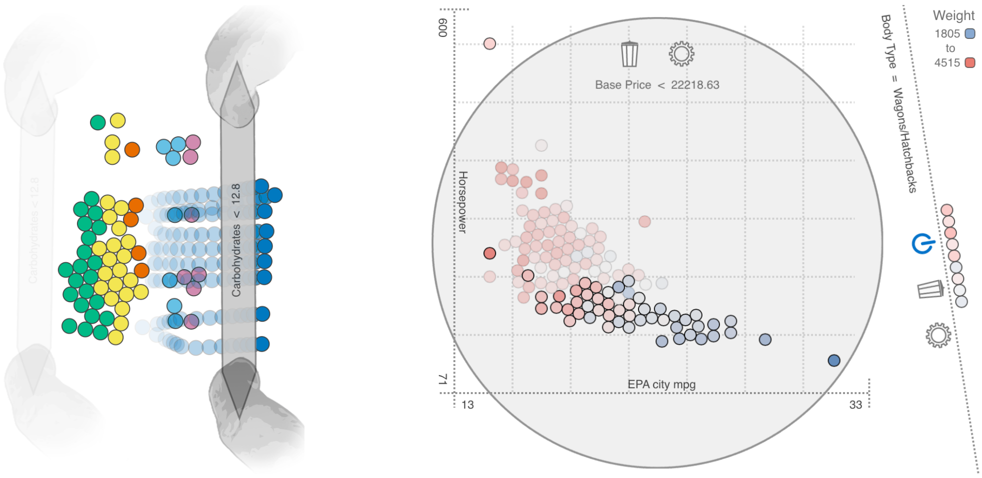



Figure 2.9: Examples of Kinetica instruments [Rzeszotarski and Kittur, 2014]. Filtering data using a semi-permeable filter (left); showing cheaper vehicles using a lens over a scatterplot where wagons are filtered out (right).

Dust & magnet is an approach for the exploration of multidimensional datasets [Yi et al., 2005], where each data point is represented as a circle (dust) and each dimension as a larger square (magnet). The interaction with dust & magnet is a straightforward metaphor of the physic world (P4, B1, B11, B16): each data point is attracted by the (weighted) magnets according to their value according to the dimension the magnet represents. Similarly as for parallel coordinates, magnets can be dragged and dropped and the data points move accordingly. Thus, each dimension of the dataset's representation becomes an instrument and interaction is performed in the 2D space (C16). Dust & magnet is both easy to learn and easy to use thanks to the consistent physical actions.

Rzeszotarski and Kittur extend the approach by providing a new set of instruments to interact with the visualization [Rzeszotarski and Kittur, 2013, 2014] (Figure 2.9). They base their design upon physics-based affordances such as magnets to employ forces and barriers to block data points, allowing for grouping points (C7). The notable advantage of Kinetica is the use of instruments directly on the 2D space where data is represented instead of widgets (C15), thus with a low indirectness (P11, P12), and a high degree of integration (P13). Moreover, the physics metaphors ensures a high degree of compatibility (P14)—what Beaudouin-Lafon would call "good instruments". The results of the user study emphasized that Kinetica is fun to use, playful, fluid and engaging (P5, B6, B7, B8, B9), indicating that the system gives a *feeling of control* and is *playful and enjoyable*. One limit of the system is that the more the instruments, the less the interface is internally consistent, resulting in a longer and steeper learning curve (C1), hindering both learning and ease of use (B12, B13).

To summarize, Dust & magnet and Kinetica highlight the tradeoff between power and simplicity. Dust & magnet is internally consistent, easy to learn and easy to use but offers limited functionalities. Kinetica introduces more effective instruments that are also easy to learn individually because metaphors of the physical world, but the interface in its whole can be difficult to learn (B12, C1). Moreover, it involves a low internal consistency and non reversible actions, lowering the ease of use of the system; but it remains playful and engaging because of the seamless, fluid interactions it provides.

#### 2.2.2.6 *Kronominer*

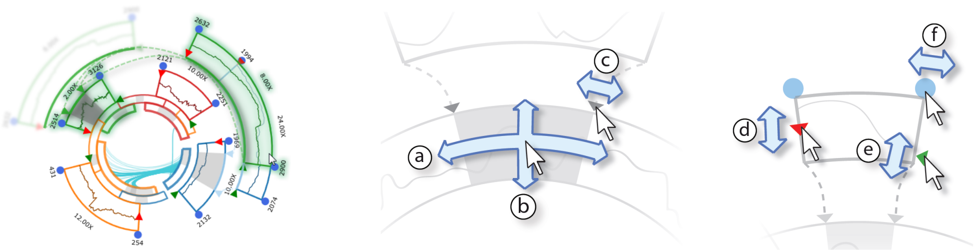



Figure 2.10: Kronominer [Zhao et al., 2011]. Left: overview; right: direct manipulation of time series segments.

Kronominer [Zhao et al., 2011] is a tool for the visual exploration of time series data (Figure 2.10) using a radial layout and supporting many interaction techniques. The main characteristic of Kronominer is that the authors paid great attention to designing interactions avoiding widgets and setting panels (C15) to interact directly with the time series and ensuring low degrees of spatial and temporal indirection (P11, P12). For example, creating a hierarchical segment is performed by brushing a region of interest and adjusting a segment is performed by dragging handles.

However, this challenge is not easy. The main drawback of the design is that it involves using different mouse buttons and several key modifiers that are not straightforward (P18)—such as right-clicking a region of interest to create a new segment, thus the interactions have a low degree of compatibility (P14). Moreover, the authors introduce *context-based interaction* exploiting the position of the cursor to determine the object of interest among a list: a single segment, a segment and its whole branch, a single ring, or the whole view. Although this strategy allows for interacting with various objects of interest, the result is an interface with a low internal consistency, similar user's action triggering different system feedback. Thus, it is difficult to learn (B12), difficult to use (B13), and its perceived quality may be low (B14). As the authors point out, *"keeping consistency while designing direct manipulation is possible although limited: there is a finite number of ways of binding operations to the combination of mouse buttons, mouse wheel and modifier keys"*.

In summary, Kronominer supports many interaction techniques by direct manipulation but at the cost of introducing both a low degree of compatibility and a low internal consistency that can make its functionalities difficult to learn and recall. That is what Jansen and Dragicevic call *non-generic* instruments [Jansen and Dragicevic, 2013]—instruments whose effects are not consistent across compatible visualizations.

#### 2.2.2.7 *ScatterDice*

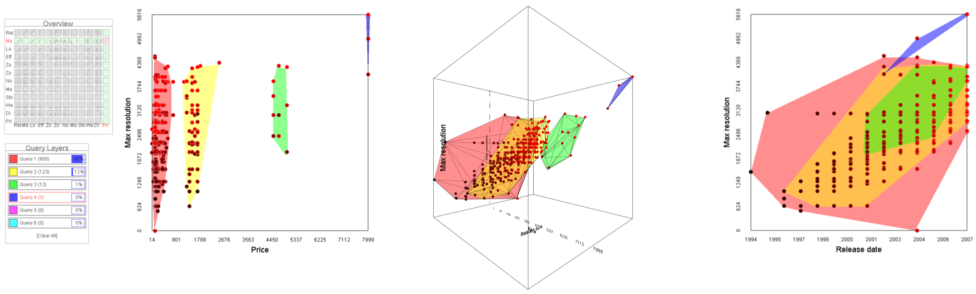



Figure 2.11: ScatterDice [Elmqvist et al., 2008a]. The axes of the query view are updated using an animated 3D rotation.

ScatterDice [Elmqvist et al., 2008a] is a visualization tool for visual exploration of multidimensional data (Figure 2.11). The main characteristic of ScatterDice is the use of an animated 3D rotation in the main view when navigating in the overview scatterplot matrix. ScatterDice also features query sculpting, a lasso brushing technique to create colored hulls in the main view. Rotating the view when navigating in the dimensions of the dataset animates the query hulls to to their new position, allowing to visually track items or groups of items.

ScatterDice is a good example of seamless and playful interaction. First, the objects are continuously represented (P1), actions are incremental and reversible (P3), and the feedback is immediate (P9, P12). Second, it gives a feeling of control to the user by minimizing both the gulf of execution (P5) and evaluation (P6); by using smooth transitions between states (P15), and by giving the illusion to manipulate the semantic objects (B9) by making the interface minimalistic (P16). The slow interaction is also beneficial to generate more insights than using automatic multidimensional clustering algorithms. Third, the interface is playful and enjoyable as it is non interfering (P10) and the 3D animation makes the interaction rewarding and playful (P17).

#### 2.2.2.8 *Bring&Go*

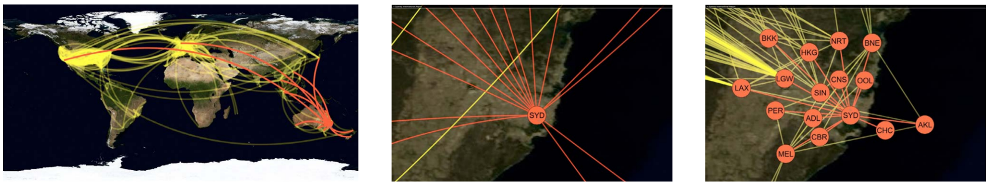



Figure 2.12: The Bring&Go navigation technique for large networks [Moscovich et al., 2009]. From left to right: overview of the network, close-up on Sydney, and the Bring&Go interaction on Sydney.

Bring&Go is a navigation techniques for navigating in large networks [Moscovich et al., 2009] without panning and zooming. Selecting a node brings all the adjacent nodes close to the selected one using animated transitions (Figure 2.12). One of the adjacent node can then be selected and the viewport is animated to the position of the new node following the link trajectory. Bring&Go is also iterative. When selecting a node, its adjacent nodes get closer. Then, a second Bring&Go can be performed on one of the adjacent nodes, and so on, until the desired final node is selected.

Bring&Go is another good example of seamless and playful interaction. As for ScatterDice, actions are incremental with immediate feedback (P1, P3, P9, P6). Using animated transitions rewards interaction (P17) and makes the interface playful at the cost of introducing temporal indirection (P12). The navigation technique was also proved to be efficient and fast for several low level navigation tasks (C2), easy to learn, easy to use and pleasant. This is mainly due to the direct manipulation of the visual objects (B9) instead of widgets (C15), making the interface invisible (P16).

Bring&Go also reduces spatial indirectness (P11) by bringing nodes closer. Interestingly, this approach allows for making visible objects that are originally invisible in the viewport. This is a way of solving challenge C6.

### 2.2.3 *Marking Menus for Visualization*


Figure 2.13: Marking menu for manipulating subgraphs in networks.

Marking menus [Kurtenbach and Buxton, 1993, 1994] have been used in the context of Information Visualization. Because marking menus are opened by interacting directly with the objects of interest, they have a low spatial indirection (P11). McGuffin and Balakrishnan designed a large number of interaction techniques for selecting and manipulating subgraphs in networks [McGuffin and Balakrishnan, 2005]. Interactions are triggered using a marking menu that pops up when selecting a node (Figure 2.13), then selecting the action, then applying the selected action by moving the mouse in the 2D space. Several interactions are congruent to the associated tasks (P2, P14, P19). For example, translating a group a nodes is performed by translating the mouse cursor in the 2D space. However, the interactions have a low temporal directness as they require 1) selecting the object of interest, then 2) selecting the command from the marking menu, and 3) performing the interaction gesture. Finally, the interface suffers from the same limitations as Kronominer as seven combinations of mouse button and keyboard press trigger interactions, requiring switching between interaction modes (P18).

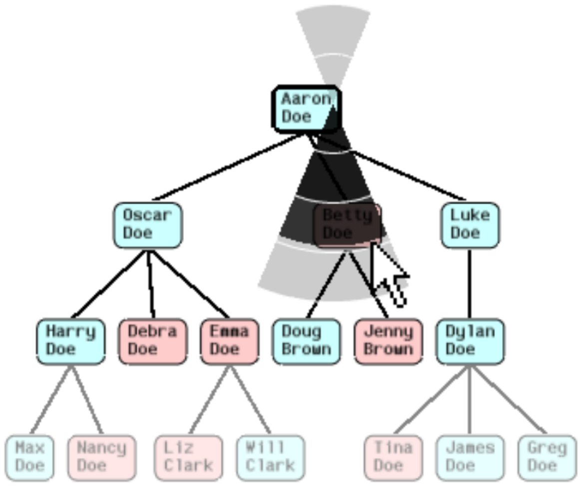



Figure 2.14: Marking menu for navigating in genealogical graphs.

Marking menus have also been used to navigate into genealogical graphs [McGuffin and Balakrishnan, 2005]. In this case, the marking menu features few commands, making the interface much simpler to use. The idea is to progressively expand and collapse a node's parents and descendants on demand. Two interactions are available, triggered with one of the two mouse buttons in a unique gesture. Pressing the left button on a node makes a marking menu appear, then dragging the mouse up and down makes the parents and descendants appear, respectively. Pressing the right buttons allows for navigating into several levels of the graph by a unique, continuous, and reversible gesture (Figure 2.14). Both interactions are performed directly on the object of interest and animated transitions (P15) provide immediate feedback (P9, P12). Moreover, the gesture direction is cognitively congruent to the task and the output result (P5, P6, P14, P19). Finally, the interactions are versatile because they are compatible with other directed graph representations by simply changing the orientation of the marking menus.

## 2.3 LESSONS AND DISCUSSION

Examining these different Infovis systems highlights several aspects and tradeoffs of direct manipulation for Infovis.

**EASE OF LEARNING VS EASE OF USE** Parallel coordinates illustrate how ease of learning and ease of use may be conflicting (C11). This technique is not easily accessible to novices: the interactions are not consistent and there are no obvious affordances to guide users, the interface being almost invisible. However, once taught and trained, experts work extremely rapidly and are highly efficient. As explained in Section 1.2 where I discarded C2 [van Wijk, 2005], learning is not the focus of this thesis. I consider that if the perceived benefits of an interaction are high enough, users will learn it. Thus, the user being on the asymptotic slope of the learning curve, I focus on ease of use.

**INTERNAL CONSISTENCY** The three types of consistency of an interface have different roles. Consistency with the world beyond computing is related to metaphors and affordances, thus to learning. Metaphors can be superficially attractive and conflict with ease of use (e. g., Kinetica). External consistency describes the similarity with other interface designs familiar to users, helping transfer of training, thus learning. It may also confine interaction design to what is known and hamper creativity and novelty, and lead to easy to learn interactive visualizations that may perform poorly [Few, 2011].

Conversely, internal consistency improves not only learning but also ease of use and perceived quality. Kronominer, Kinetica and complex marking menus illustrate how a low internal consistency can make the interface difficult to both learn and use, and raise the tradeoff between power and simplicity of an interface. Usually, the more powerful the interface, the more complex and heterogeneous the interface. Chapter 4 and Chapter 5 show examples of powerful yet simple interfaces, where a unique interaction allows for performing efficiently a large number of tasks. Internal consistency also allows for a coherent interaction model, especially for feature-rich applications [Chevalier et al., 2012]. Moreover, internal consistency and external consistency being conflicting [Grudin, 1989], focusing on internal consistency is another reason for discarding external consistency. Chapter 6 illustrates the design of an interface based on a unique interaction metaphor to accomplish a wide range of tasks.

**CONGRUENT INTERACTIONS** Dynamic queries highlight the difficulty of designing congruent interaction. Congruent interactions lead to easier to use applications by providing a natural mapping to the task, reducing the semantic indirectness both in the gulf of execution and in the gulf of evaluation.

A crucial challenge is the design of 2D interactions for multidimensional information spaces using a two-dimensional input device. Dust & Magnets and parallel/star coordinates feature examples of interactions to explore multidimensional datasets. In Chapter 4 and Chapter 5, I investigate direct manipulation seamless interaction techniques to navigate in the temporal dimension of two-dimensional data graphics.

Surrogate objects solve some of the direct manipulation challenges but at the cost of introducing spatial indirectness. However, being too (spatially) direct is not always for the best. For example, drawing on interactive tables involves occlusion of the drawing but providing a feedback on a spatially distant screen solves this issue. Other trivial examples include the switch for a light and the steering wheels for the wheels of a car, where spatial directness is clearly undesirable. These examples show that spatial indirectness can be balanced by brain plasticity, and it only involves learning. Introducing spatial indirection can result in easier to use interfaces, and that is the approach I chose in Chapter 6 where instruments have a spatial directness of 1/2 (according to only one dimension of the 2D space).

EASE OF USE VS SEAMLESS, FLUID INTERACTION The reorderable matrix emphasizes the need for fluid interaction to steer algorithms to accelerate the process and be more efficient while generating insights. Interactions for human-steered algorithms are explored in Chapter 6. ScatterDice and Bring&Go also illustrate the tradeoff between immediate feedback and increased ease of use using smooth transitions to feel in control. Marking menus for navigating in genealogical graphs shows that fluid interaction can be compatible with an easy to use interface.

TARGETING UNSEEN OBJECTS The challenge of targeting unseen objects has two aspects. Surrogate objects allows for selecting unseen properties of objects by providing more information on the surrogate than can be shown on the associated object. Bring&Go allows for making visible objects that are not in the viewport, without panning and zooming, i. e. without altering the viewport. These are two different and important challenges. In Chapter 3, I propose an interaction technique to reveal unseen properties of time series. In Chapter 5, I explore ways of navigating in large visualizations where objects of interest can go beyond the viewport.

DIRECTNESS CONTINUUM All these examples show the non-binary type of direct manipulation: there is a multidimensional continuum from indirect to direct manipulation where dimensions are the listed principles. WIMP interfaces are certainly more direct than command-based interfaces but their degree of directness is low. Hutchins et al. already asserted that *"the notion of 'direct manipulation' is not a unitary concept, nor even something that can be quantified in itself. It is an orienting notion. 'Directness' is an impression or a feeling about an interface."* [Hutchins et al., 1985], and several other researchers stressed this notion of continuum (e. g., [Jansen and Dragicevic, 2013; Lee et al., 2012; Tweedie, 1997].

### 2.3.1 *Darwinian interaction selection*

More generally, there is no predictive theoretical ground to assess what makes a good interaction for Infovis. If science is considered as a Darwinian selection process among competing theories [Popper, 1959], the science of interaction needs opportunistic improvements to accelerate the evolution process [Fekete et al., 2008]. I embrace this approach, given that once a new interaction paradigm is learnt, used and integrated by the user, it becomes a direct cognitive connection [Clark, 1997].

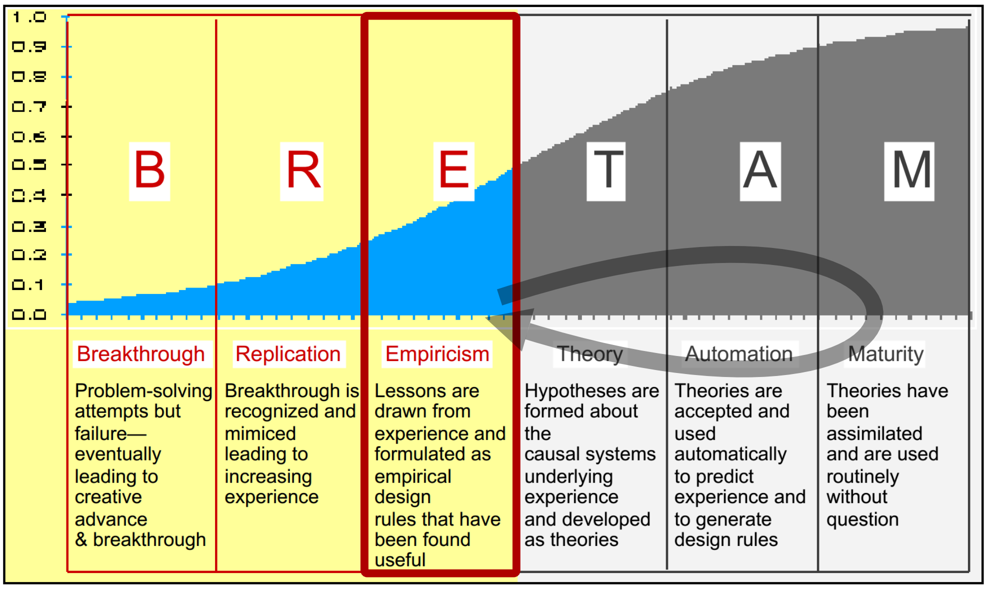



Figure 2.15: The BRETAM model showing the six qualitative changes along a technological learning curve, adapted from [Gaines, 1991]. This thesis focuses on the **E**mpiricism stage.

The BRETAM model [Gaines, 1991], illustrated Figure 2.15, shows the different steps of the long process leading to a technology maturity. Considering the breakthrough of the WIMP paradigm that was followed by a large number of replications, interaction for Infovis is not yet at its **M**aturity phase and a lot remains to be explored. Principles derived from direct manipulation such as the instrumental interaction dimensions are empirical rules. In the case of Infovis interaction techniques, there is an iterative process going back and forth between **E**mpiricism and **M**aturity. The **T**heory and **A**utomation stages are not well explored and techniques can be used routinely while not **M**ature yet. Thus, the field of Infovis is still at the **E**mpiricism stage where new opportunistic interaction techniques need to be created based on **E**mpirical rules before formalizing a theory of interaction for Infovis.

This raises the problem of characterizing the fields of HCI and Infovis. Many advances in both fields have reached their maturity and are used routinely based on empiricism while the theoretical ground is in its infancy. Thus, the fields of HCI and Infovis seem to be closer to humanities and social science where ground theories are rare, than to natural and formal sciences. Because there is no ground theory to explore the space of possibilities completely, the solution is to explore step by step, empirically and opportunistically.

Hence, this thesis focuses on this **E**mpiricism stage by exploring opportunistic interactions for existing visual representation techniques. As for natural Darwinian selection process, not all interaction techniques will pass the test of time but the gathered lessons and design guidelines are one more step towards a theory of interaction for Infovis.

# 3 IMPROVING THE EFFICIENCY OF AN EXISTING VISUAL REPRESENTATION

This chapter is based on previous publications [4], [10]. Thus any use of "we" in this chapter refers to Charles Perin, Frédéric Vernier, and Jean-Daniel Fekete.

Dix and Ellis argues that *"The heart of modern visualization techniques is interaction and [...] interaction can be applied to any representation however simple."* [Dix and Ellis, 1998]. This chapter follows this statement and aims at providing a simple example illustrating how an existing visualization technique can be greatly enhanced by adding simple yet powerful interactions to it.

The question we address in this chapter is a particular case of the general question of the thesis:

- ▷ How to improve the efficiency of Horizon Graphs through effective interactions, and what are the benefits of such interactions?

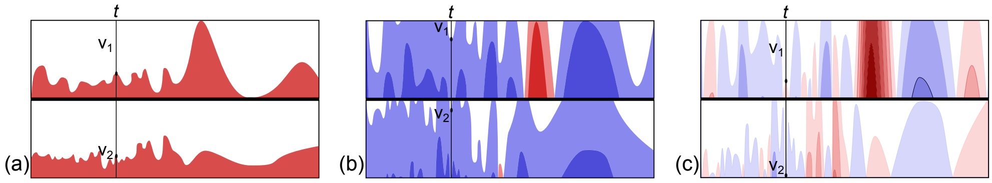



Figure 3.1: Two time series visualized in parallel using RLC, HG and IHG. The degree of difficulty when determining which of the series has the highest value at point t (marked by a vertical black line) is different for each technique: (a) Using RLC, it is very difficult to compare $v_1$ and $v_2$. (b) Using HG with standard baseline at half the $y$ axis and with two bands, we can barely see that $v_1 > v_2$: since both charts are blue at that point (i. e. under the baseline), the highest value is the lowest blue one. (c) Using IHG, setting the baseline at 28% of the range of values and a zoom factor of 6, it is clear that $v_1 > v_2$: only $v_1$ is shown in red, i. e. above the baseline.

*Time series*—sets of quantitative values changing over time—are predominant in a wide range of domains such as finance (e. g., stock prices) and sciences (e. g., climate measurements, network logs, medicine).

Line charts are one of the simplest ways to represent time series, and one of the most frequently used statistical data graphics [Cleveland, 1993]. However, using line charts to visualize multiple time series can be difficult because the limited vertical screen resolution can result in high visual clutter.

We introduce Interactive Horizon Graphs (IHG), an interactive technique for visualizing multiple time series. IHG are inspired by pan and zoom techniques and unify Reduced Line Charts (RLC) and Horizon Graphs (HG), two of the most effective techniques for visualizing multiple time series. We designed IHG to increase the number of time series one can monitor and explore efficiently. Datasets involving large numbers of time series such as stocks or medical monitoring are frequent and important [Hochheiser and Shneiderman, 2004]. We evaluate the benefits of our contribution for standard tasks on time series visualizations. While the related work has used generated time series with clear landmarks for evaluation, we used a non-synthetic Large scale and Small scale Variations dataset (LSV) adapted to multi-resolution visualization techniques.

Under these conditions, we obtained results that are different from those in previous work [Heer et al., 2009; Javed et al., 2010] (performances are better for HG than for RLC) and found that IHG outperform both RLC and HG when the number of time series is higher than 8.

## 3.1 RELATED WORK

Since line charts have become widespread [Playfair, 1786], visualization of time series has been an active research topic, moving from paper-based representations to interactive visualizations. Many design considerations exist for displaying data in the form of charts (e. g., [Bertin, 1983; Cleveland, 1985; Tufte, 1986]) and for the comparison of graphical visualization techniques (e. g., [Peterson and Schramm, 1954; Simkin and Hastie, 1987]). For relevant surveys see [Aigner et al., 2011; Silva and Catarci, 2000].

### 3.1.1 *Visualization Of Multiple Time Series*

Visualizing multiple time series in a small space (where the vertical resolution is smaller than the series variations one may be looking for) has led to techniques that use space-filling [Wattenberg, 2005] and multi-resolution representations [Lam et al., 2007].

Javed et al. classified visualization techniques for multiple time series into two categories [Javed et al., 2010]. In *shared-space* techniques, time series are overlaid in the same space (e. g., *line graphs* [Playfair, 1786], *braided graphs* [Javed et al., 2010], *stacked graphs* [Byron and Wattenberg, 2008]). In *split-space* techniques, the space is divided (usually horizontally) by the number of time series and each one occupies its own reduced space (e. g., RLC [Tufte, 1986], HG [Few, 2008; Reijner, 2008]). Shared-space techniques can support only a limited number of time series (considering more than four involves too much visual clutter [Javed et al., 2010]). Because we focus on large numbers of time series, we only consider split-space techniques. Also, while most of prior techniques are static, we focus on evaluating the benefits of adding interaction.

### 3.1.2 *Reduced Line Charts (RLC)*

RLC are small multiples for time series using line charts. To perform comparison tasks on different RLC, they must all share the same range of values (Figure 3.1(a)).

### 3.1.3 *Horizon Graphs (HG)*

HG is a recent split-space technique intended to display a large number of time series. It was originally introduced under the name "two-tone pseudocoloring" [Saito et al., 2005] and was later developed by the company Panopticon under the name "horizon graph" [Few, 2008; Reijner, 2008]. This technique uses two parameters: the number of bands $b$ and the value of the baseline $y_b$ separating the chart horizontally into positive and negative values.

Figure 3.2 illustrates the construction of HG from a line chart centered around a baseline. First, the values are colored according to their position relative to the baseline (3.2(a)). Next, the line chart is horizontally split into

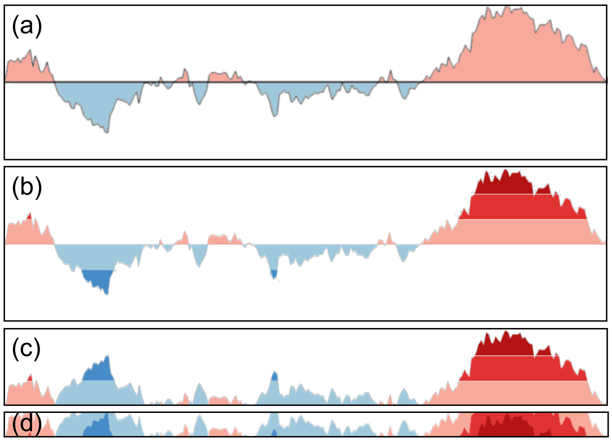



Figure 3.2: The construction of a *Horizon Graph* with 3 bands, adapted from [Few, 2008; Javed et al., 2010]. (a) Values are colored (blue and red) according to their value compared to the baseline: blue below and red above. (b) The chart is split in 3 bands (3 reds and 3 blues). (c) Values below the baseline are mirrored. (d) The bands are wrapped.

uniformly-sized bands and their saturation is adjusted based on each band's proximity to the baseline (3.2(b)). The bands below the baseline are then reflected above the baseline (3.2(c)), so that the height of the chart becomes half of what it was originally. Finally, the different bands are layered on top of one another (3.2(d)), reducing the final chart height to $h/(2 \times b)$, where $h$ is the original height of the chart and $b$ is the number of bands. Using HG, data values are represented not only by their vertical height, but also by their color saturation and hue. For instance, the global maximum of a time series is the highest of the darkest red values. Figure 3.1(b) illustrates two HG in parallel.

Heer et al. [Heer et al., 2009] evaluated the use of HG focusing on how chart-reading performance changed using different parameters. They provide some recommendations, such as the optimal chart height and the number of bands which should be used. They limited their study to two simultaneous time series and the number of bands to four. Javed et al. [Javed et al., 2010] compared HG with other visualization techniques for higher numbers of time series. They limited the HG parameters to those recommended by Heer et al. and did not highlight any considerable advantage of the technique. In particular, they did not find critical differences between RLC and HG. However, they found that the number of time series seriously impacted the visual clutter and played a very important role in the performance of the visualization techniques. In their experiments, both pieces of prior work used synthetic data that included clear landmarks, which may have aided visual search tasks. As HG is a multi-resolution visualization technique, we can expect different results for the more difficult LSV datasets.

### 3.1.4 *Large Scale and Small Scale Variations Datasets*

Techniques such as stack zooming [Javed and Elmqvist, 2010] and dual-scale data charts [Isenberg et al., 2011] use focus+context [Cockburn et al., 2009] techniques to visualize time series data containing regions with high variations. These techniques magnify and increase the readability of regions of interest by modifying the $x$ axis (time scale), but not the $y$ axis (value scale). We only found one article [Lam et al., 2007] that explored LSV datasets exhibiting both large and small variations visible at low and high resolutions. However, time series with these properties are common—for example, one may observe the temperature of a city along one year according to different variation scales: large (seasonal), medium (daily), small (hourly).

According to Bertin, the scale of time series with small variations must be adjusted to get closer to the optimum angular legibility, which is 70 degrees [Bertin, 1983] and multi-scale banking to 45 degrees has been extensively studied in order to improve the graphical perception of time series [Cleveland and McGill, 1987; Heer and Agrawala, 2006; Talbot et al., 2012]. While several tasks can be accomplished on time series where each chart has its own $y$ axis (e. g., compare the trend of two time series during a period of time), related work [Few, 2008; Heer et al., 2009; Javed et al., 2010] suggests that the best configuration for multiple time series consists of sharing the same $y$ axis, i. e. using the same scale of values and baseline.

### 3.1.5 *Tasks on multiple time series*

Time series visualization techniques have been studied extensively and prior work has evaluated their use for a variety of different tasks. According to Andrienko and Andrienko [Andrienko and Andrienko, 2005], tasks on multiple time series can be of two types: *elementary* (about individual data elements) or *synoptic* (about a set of values). For each type, the tasks can be *direct/inverse comparison* tasks or *relation-seeking* tasks. The closest study to our work, that inspired us [Javed et al., 2010], evaluated RLC and HG considering three tasks: *Maximum*, *Discriminate* and *Slope*.

#### 3.1.5.1 *Find the Maximum (*Max*)*

*Max* is an elementary task for direct comparison. It consists of determining which of several time series has the highest (or lowest) value at a shared marked point [Javed et al., 2010; Lam et al., 2007]. Javed et al. compared RLC and HG using this task for 2,4 and 8 time series. Their study revealed that RLC were faster than HG but they did not find any significant result for Correctness.

*Max* is, for instance, executed to find the hottest city in a country for a given date. This task can be very easy to achieve if there are clear differences between the cities but becomes difficult when both the differences and the vertical resolution are small. Figure 3.1(a) and Figure 3.1(b) illustrate *Max* using RLC and HG, respectively. This example highlights the difficulty of such a simple task using LSV datasets.

#### 3.1.5.2 *Discriminate (*Disc*)*

*Disc* is an elementary task for relation-seeking, similar to *Max*. However, instead of having to find the highest value at a marked point t shared by all the time series, each time series has its own marked point. *Disc* is more difficult than *Max* [Heer et al., 2009; Javed et al., 2010; Simkin and Hastie, 1987] and HG has been evaluated for this task in two recent studies:

Heer et al. have studied the impact of the number of bands in HG [Heer et al., 2009] for *Disc*. They found that time and error increased with the number of bands. However, these results were obtained for value estimation tasks and they aptly noticed that these increases were due to the mental math implied.

For their *Disc* task, Javed et al. asked subjects to answer by selecting the time series with the highest value, rather than by estimating the highest value. They did not find any significant difference in terms of Correctness or Time between RLC and HG for *Disc*.

#### 3.1.5.3 *Evaluate the Slope*

*Slope* is a synoptic task for pattern comparison proposed by Beattie and Jones [Beattie and Jones, 2002]. It consists of determining which time series has the highest increase during a given time period. For this task, Javed et al. found no significant results for Correctness and found HG to be slower than RLC [Javed et al., 2010]. We believe that these results were also due to the synthetic dataset they used and we expect different results from a more difficult dataset.

In conclusion, previous studies on multiple time series had two main limitations: they only studied small numbers of time series ($\leqslant 8$), when much larger numbers are available in popular datasets, and used synthetic datasets, with features simpler than those typically found in these popular datasets.

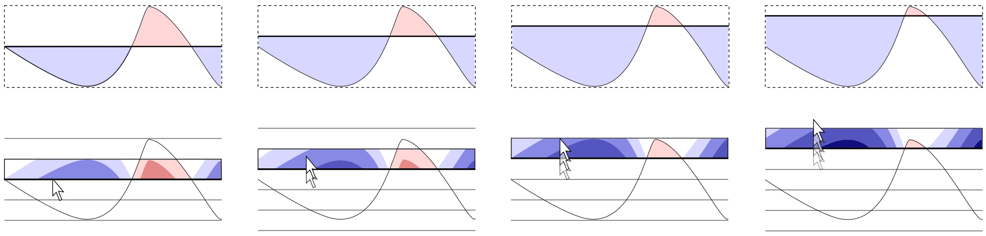


Figure 3.3: Baseline panning: The bottom charts represent the view of the time series using IHG for 4 different values of $y_b$ overlaying the original line chart (for a constant zoom factor $z = 2$). Dragging upwards the mouse with the right button pressed increases the value of $y_b$ (sequence from left to right) and values going under $y_b$ become blue. The original line chart is presented above each step for better understanding.

## 3.2 INTERACTIVE HORIZON GRAPHS

Interactive Horizon Graphs (IHG) unify RLC and HG by introducing interactive techniques to control the baseline position and the zoom factor applied to values. Interaction is meant to allow HG to remain effective even while exploring larger numbers of time series. *Baseline panning* and *value zooming* can be seen as variants of the commonly used pan and zoom interaction techniques [Bederson and Hollan, 1994]—the baseline is controlled through a variant of panning and the number of bands through a variant of zooming. Thus, the pan and zoom interaction techniques are related to the $y$ axis of the visualization instead of the $x$ axis as described in [Isenberg et al., 2011]. We detail our interaction techniques in the following subsections.

### 3.2.1 *Baseline Panning*

Baseline panning allows users to interactively move the baseline along the $y$ axis—in our implementation, this is achieved by dragging the mouse up-/down with the right button pressed. Note that baseline panning does not change the positions on the $x$ axis at all, unlike regular panning, and it does not change the height of the chart. The user's interaction with a single chart simultaneously changes the baselines on all small multiples. Because the baseline is always at the bottom of the chart, it does not move in response to the interaction. Rather, the series appear to shift up or down as the baseline changes and colors change as points in the series move from one band to the next (Figure 3.3).

Interactively changing the baseline overcomes a limitation of the fixed baseline used in traditional HG—because pre-attentive color perception (distinguishing between red and blue) is only effective for values around the baseline, points far from the baseline are more difficult to discriminate. Baseline panning allows a user to make transitions around a value of interest more salient. This can be particularly valuable if one is interested in identifying deviations from a specific baseline—for comparing the in body temperature for a patient against the patient's expected value. Meanwhile, finding a maximum value becomes a comparison of intensity of red plus height ($y$)

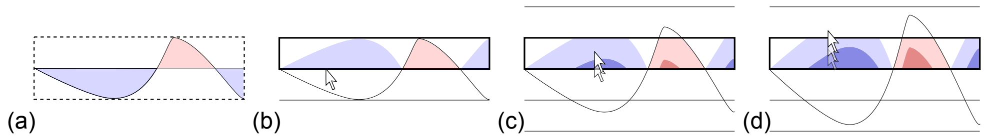



Figure 3.4: Value zooming: (a) From a standard mirrored line chart, the zoom value $z$ is progressively increased by dragging upwards the mouse with the left button pressed (for a constant baseline $y_b = \frac{y_M - y_m}{2}$): (b) $z = 1.0$, (c) $z = 1.35$, (d) $z = 1.70$. Values reaching the top of the $y$ axis appear at the bottom of the chart, with a more saturated hue. The original chart (deformed according to $z$) is overlaid for each step, for better understanding.

estimation (first search the most red-saturated areas, then find the highest value which belongs to one of these areas).

For RLC, HG, and IHG, all the charts have the same range of values for the $y$ axis: $[y_m, y_M]$, with $y_m$ and $y_M$ being the minimum and the maximum values in the visualized dataset. The three techniques have different values for the baseline $y_b$: $y_{b_{RLC}} = y_m$ (the baseline is always at the bottom of the chart), $y_{b_{HG}} = \frac{y_M - y_m}{2}$ (the baseline crosses the $y$ axis at its middle point), and $y_{b_{IHG}} \in [y_m, y_M]$ (the baseline can take any value in the range of values).

### 3.2.2 *Value Zooming*

Value zooming allows users to specify the zoom factor using a continuous interaction—in our case, dragging the mouse up/down with the left button pressed. Note that value zooming does not change the scale of the $x$ axis, unlike regular zooming, and it does not change the height the of chart, since the values will wrap around the lower border of the chart.

HG use a discrete number of bands, so changing from 2 to 3 bands triggers a sudden transition. The continuous interaction we propose prevents this abrupt change, resulting in a smooth and continuous zoom, as seen in the three zoom levels shown in Figure 3.4. The chart can be seen as if drawn on a tall sheet of paper which is wrapped around its baseline according to the zoom factor: when the shape of the chart reaches the top of the $y$ axis, it is cut and appears at the bottom of the $y$ axis, with a more saturated hue. The appropriate zoom factor depends on the scale of the variations one wants to analyze: observing small variations will result in a high zoom value and large variations in a low zoom value. Using Heer et al.'s terminology [Heer et al., 2009], our zooming implementation keeps the height of the horizon graph fixed but increases the virtual resolution of the underlying chart.

We were interested in observing how users would adapt and understand this unusual metaphor. We believe that this interactive virtual resolution control provided by our zoom can be easily understood thanks to the paper-wrapping metaphor, and that this interaction can lead to substantially higher numbers of bands than the recommended two. However, increasing the number of bands makes it more difficult for users to discriminate the different color intensities. This trade off rests in the user's hands, according to the task and/or the data. While standard zooming techniques consist of focusing on a specific area and losing context information, our zooming implementation for IHG preserves both the visibility of the context and the details of small variations around the baseline.

The range $r_i$ of each band $b_i$ is computed differently for HG and IHG because of the different values for $y_b$ and because HG use a discrete number of bands $b$, while IHG use a continuous zoom value $z$:

$$r_i = [y_b + i\frac{h}{2K}, y_b + (i+1)\frac{h}{2K}], \quad \text{with}$$

$$\text{HG}\begin{cases} i \in [-b, b[ \\ h = y_M - y_m \\ K = b \end{cases}$$

$$\text{IHG}\begin{cases} i \in [-\lceil z \rceil, \lceil z \rceil[ \\ h = \max(|y_b - y_m|, |y_b - y_M|) \\ K = z \end{cases}$$

### 3.2.3 *Combination Of Pan And Zoom*

The technique we provide never leads to loss of information thanks to the HG properties. Moreover, for both our pan and zoom interaction techniques, the visual feedback is different from a standard pan and zoom along the $x$ axis and results in user-controlled transitions instead of sudden changes.

To illustrate the effectiveness of our technique, let's consider the basic task of finding the global maximum over multiple time series. This task is accomplished in two steps: first, the baseline is set at $y_M$ so that all the values are colored blue. Then, the value of the baseline is progressively decreased by the user until red values appear in one or several charts. The global maximum belongs to one of these charts. If two or more time series turn red for the same value of the baseline, the user will zoom in to enlarge these areas and the differences in magnitude will be visible.

Another typical use of our technique consists of locking the pan to a reference value of interest and zooming to highlight the differences with the other values. This case is illustrated in Figure 3.5: let's consider a time series with small variations around a specific value except during a period of time containing higher values, resulting in a high bump (Figure 3.5(a)). Using the recommended parameters ($z = 2.0$, $y_b = \frac{y_M - y_m}{2}$, Figure 3.5(b)) slightly increases the small variations but the baseline separating the chart in two brings no interesting information because the value of interest is not near $y_b$ and HG is not adapted to such a case. With a well-chosen value for $y_b$ (Figure 3.5(c)) one can focus on the value of interest. Still, the differences between values are difficult to estimate. Combining pan and zoom ($z = 8.5$, $y_b = 0.08 \times (y_M - y_m)$, Figure 3.5(d)) makes the small variations easy to

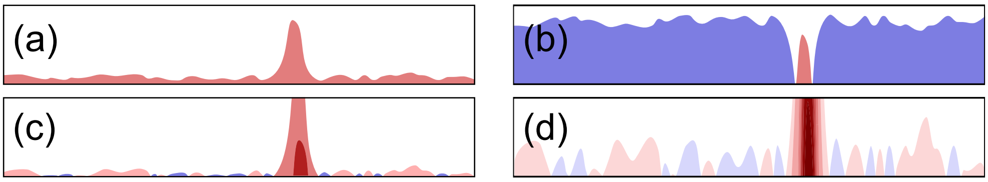



Figure 3.5: Four views of a time series illustrating the importance of the interactive settings of the baseline value $y_b$ and the zoom factor $z$.
(a) $y_b = y_m$, $z = 1.0$; (b) $y_b = \frac{y_M - y_m}{2}$, $z = 2.0$;
(c) $y_b = 0.08(y_M - y_m)$, $z = 2.0$; (d) $y_b = 0.08(y_M - y_m)$, $z = 8.5$.

read and compare. Furthermore, Figure 3.1(c) illustrates how *Max* can be easily accomplished using IHG in comparison to RLC and HG. These examples illustrate the importance of properly setting the number of bands and the value of the baseline. Those settings need to be interactively set because they depend on which part of the chart and on which type of variations (large or small) one is interested in.

Finally, we designed our pan and zoom interaction techniques keeping real-world scenarios in mind. For instance, baseline panning would let a doctor specify the base value for the body temperature of patients according to their health. The continuous zoom provides an effective way of exploring the temperatures of a city during one year; according to the zoom factor, seasonal, daily, or hourly variations may be observed.

## 3.3 USER STUDY

We designed an experiment to determine the usefulness of adding interactivity to HG. In the study we asked users to examine LSV datasets and perform three kinds of tasks using RLC, HG, and IHG. To quantify the impact of each approach, we measured the *Time*, *Correctness*, and *Error magnitude* for each visualization technique.

### 3.3.1 *Data*

We used several datasets, including unemployment rates and temperatures, during our pilot studies. However, for the main experiment we chose real-world data from Google Finance [Google Finance]. We used the stock market history during February 2012 from 182 banks with no missing data for that period. We chose these datasets because they are LSV time series that evolve in a close range, making it necessary to use a common scale for all visualized charts. Because LSV time series have different levels of detail, we expected that HG would outperform RLC and that we would be able to differentiate HG and IHG, since both are multi-resolution visualization techniques.

### 3.3.2 *Hypotheses*

Our hypotheses for this experiment were as follows:

**H1** *The benefits in terms of Time, Correctness and Error of IHG compared to RLC and HG will increase with the number of time series .* This hypothesis is based on the intuition that the task becomes more difficult with larger numbers of time series but that interaction will help deal with the increasing scale. To test this hypothesis, we designed variants of the task using 2, 8, and 32 time series. We also predicted that the greater the number of time series, the less efficient RLC will be.

**H2** *IHG will be faster for all the tasks.*

**H3** *HG with its recommended parameters ($y_b = \frac{y_M - y_m}{2}$ and $b = 2$) will be less efficient than IHG for LSV time series.*

### 3.3.3 *Experimental Factors*

We describe in the next subsections our experimental factors: *visualization technique*, number of time series $N$ and *task*.

#### 3.3.3.1 *Visualization Techniques*

Across all three visualization conditions (RLC, HG and IHG), each of the charts was given the same height and all charts shared the same value range and the same baseline value. Based on previous work, we chose a constant height of 24 pixels for the charts, regardless of the number of displayed time series. Heer et al. found this height to be optimal for both RLC and 1-band mirrored

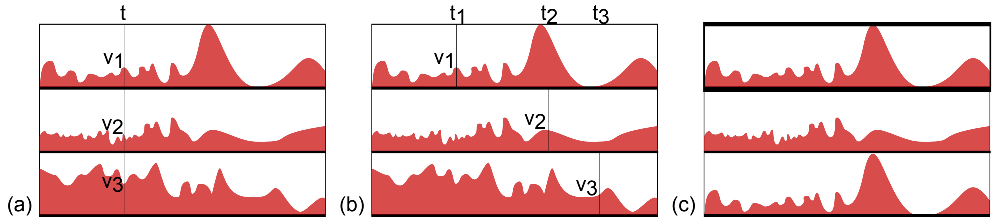



Figure 3.6: Narrower visuals of the three tasks. (a) *Max*: select the time series having the highest value at $t$. *Disc*: select the time series $i$ having the highest value at $t_i$. *Same*: select the time series $i, i > 1$, being the copy of the reference time series $i$.

HG [Heer et al., 2009], and using this size allows us to compare our results to theirs. We also made several specific choices in the design of each condition:

**RLC**: for consistency with HG and IHG, the charts were filled in with the color corresponding to values above the baseline. Although the data values were not all positive, the baseline was at the overall dataset minimum value $y_m$.

**HG**: we reversed the meaning of red/blue in our color map because, during the experiment design and pilots, we tested datasets with temperatures that are usually encoded using blue for cold and red for warm. This flipping of colors does not bias the experiment since the coding is consistent over the three techniques. We used the recommended values $y_b = \frac{y_M - y_m}{2}$ and $b = 2$.

**IHG**: to facilitate learning, we chose the value of the baseline and the zoom factor at the initial stage to be the same as the ones for RLC , i. e. $y_m$ and 1.0, respectively. The color coding was identical to the one used for HG. During the experiment, the value of the baseline and zoom factor were displayed.

#### 3.3.3.2 *Numbers Of Time Series (N)*

The related work on graphical perception of multiple time series often considered only two time series at a time [Heer et al., 2009; Simkin and Hastie, 1987]. More recently, Javed et al. compared different visualization techniques with higher values for N: their main study dealt with 2 to 8 time series and their follow-up included up to 16 time series [Javed et al., 2010]. We considered sets of *N=2* and *N=8* time series so that we could compare our results against prior work. In addition, because one of our goals was to deal with larger numbers of time series and test the scalability of split-space techniques, we also considered sets of *N=32* series.

#### 3.3.3.3 *Tasks*

Based on the task taxonomy for time series developed by Andrienko and Andrienko [Aigner et al., 2011; Andrienko and Andrienko, 2005], we chose one elementary task for direct comparison (*Max*), one elementary task for relation-seeking (*Disc*), and one synoptic task for relation-seeking (*Same*) (Figure 3.6).

The *Find the same* (*Same*) task is a variant of the Andrienko and Andrienko's *Slope* task. Users are asked to select the time series that is exactly the same as a specified *reference* time series. We chose this alternative because of the very high difficulty in discerning the slope of time series using RLC with LSV datasets. Our selection of this particular set of tasks was motivated by our pilot studies and was designed to allow us to compare our results against prior work.

We also discarded several other tasks from our experiment based on the results of pilot studies. For example, we did not ask users to find the global maximum across all the time series because IHG were clearly better for this task than the two other techniques in terms of Correctness and Time. Furthermore, automatic techniques would outperform any interactive technique for this kind of basic task.

FIND THE MAXIMUM (MAX): We chose to have more control on the task than previous experiments to adapt it to LSV time series. A *reference* time series is randomly picked from the dataset and assigned a random position in the display order. This *reference* is marked at a random point in time $t$. Its associated value is $V_t$. The other time series are then selected in the dataset if they satisfy the following condition: being $v_t$ the value of each additional time series at $t$, the time series is said to be *comparable* with the *reference* if:

$$\begin{cases} V_t - v_t & > & 2\% \times (y_M - y_m) \\ V_t - v_t & < & 10\% \times (y_M - y_m) \end{cases}$$

By imposing these conditions, the minimum visual difference between the *reference* value and the remaining time series values at the shared marked point *t* is in the range $[0.5, 2.5]$ pixels for the RLC technique. For HG and IHG , the difference in pixels is proportional to the virtual resolution [Heer et al., 2009], i. e. the number of bands.

DISCRIMINATE (DISC): The time series are selected in the same way as in *Max* but each has its own random time-point $t$.

FIND THE SAME (SAME): There is one more time series displayed for this task than for the two others (the *reference*).

Because we are focused on assessing visual perception of time series, we did not include additional features such as sorting or highlighting maximum values that might help users perform operations like *Max* and *Disc*. As in Javed et al.'s study [Javed et al., 2010] we provided no scale or tick marks and displayed no numerical values. Participants were only able to analyze the shape and colors of the time series. Note that these tasks are very difficult to perform if the differences in magnitude between the values are small, which is the case for LSV datasets.

### 3.3.4 *Overall Experiment Design*

The dependent variables we measured are *Time* (continuous) and *Correctness* (binary). Because Correctness does not capture the error's magnitude, for *Max* and *Disc* we also measured the *Error* (continuous), which is defined as $\frac{100 \times e}{(e_M - e_m)}$, where $e$ is the absolute error measured, and $e_M$ and $e_m$ are the maximum and minimum possible errors. Error expresses the difference in percentage between the correct maximum value and the value chosen by the user. For *Same*, this additional measure has no meaning unless we subjectively define a similarity measure. Therefore, we only recorded the Correctness of the answer in *Same*. For IHG, we also measured how long each participant took to perform the pan and the zoom interactions, as well as their values at the end of each trial. Each participant performed four trials per *technique* $\times$ *task* $\times$ $N$ combination. The order of *technique* and *task* was counterbalanced using a Latin square to minimize learning effects.

Because the difficulty of the task is highly correlated with the number of time series [Javed et al., 2010], the order of $N$ was gradually increased instead of being randomized (first 2, then 8, and finally 32). In summary, the design included $(3 \times \mathtt{techniques}) \times (3 \times \mathtt{tasks}) \times (3 \times N) \times (4 \times \mathtt{trials}) = 108$ trials per participant. For each, the time series were randomly selected in the dataset. The experimental session lasted about 45 minutes in total.

Participants finished the trials for a particular technique, separated into task blocks, before moving on to another one. Each time a new task began (three times for each technique), participants went through a short training for that block. This training consisted in a reminder of the task and four training trials, not limited in time to let participants establish their strategy for the task. During the training as well as the actual trials, participants received feedback as to whether their answer was correct or not. There were told that the Correctness of the answer was more important than the Time.

### 3.3.5 *Participants*

Nine participants (7 males, 2 females) were recruited from our research institute. Participants ranged from 23-36 years in age (mean 27, median 26), had normal or corrected-to-normal vision and were not color blind. Participants were all volunteers and were not paid for their participation in the experiment. All the participants (students as well as non-students) had a background in computer science and good chart reading skills. Six participants had already heard of RLC and only one knew HG.

### 3.3.6 *Procedure*

The participants watched a short introductory video explaining the RLC and HG techniques and illustrating the possibility of modifying the baseline to separate the values below and above it by coloring a standard line graph. They sat in front of a 19 inch LCD monitor (1280x1024 pixels) at a distance of approximately 50 cm and used only the mouse during the experiment.

Table 3.1: Percent of participants using no interaction, only the pan, only the zoom, and both interaction by $N$, all tasks combined.

| N | **None** | **Only Pan** | **Only Zoom** | **Both** |
|---|---|---|---|---|
| 2 | 46.7 | 6.6 | 10 | 36.7 |
| 8 | 3.3 | 6.7 | 18 | 71.7 |
| 32 | 3.3 | 0 | 10 | 86.7 |

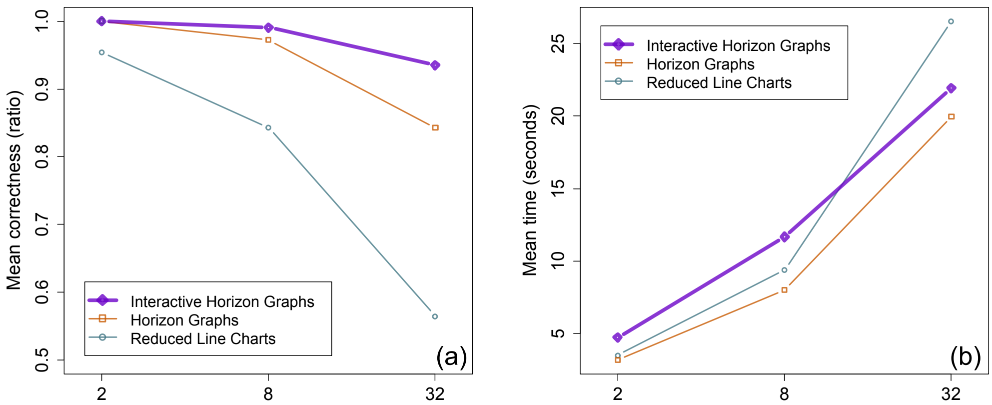



Figure 3.7: (a) *Correctness* and (b) *completion time* plots for each technique for the overall study (all tasks combined) as a function of N.

To select an answer time series, they had to double-click on it. To avoid accidental clicks, after having selected the time series, a dialog asked them to confirm their choice while the time kept running. This interaction was the only one available for RLC and HG. For IHG, pan and zoom were provided using the mouse by dragging vertically anywhere on the screen with one of the two mouse buttons pressed. The left button triggered the zoom and the right button the pan. Participants were able to practice until they understood the interface well. After each task and for each technique, participants were asked to give a score for *difficulty* and describe the strategy they used.

## 3.4 RESULTS

All data were analyzed using repeated ANOVA measures. We applied a log transform to the measures of Time to obtain a quasi-normal distribution. Pairwise t-tests were done with the Bonferroni adjustments. Effect sizes were computed using the unbiased estimate of Cohen's d [Cohen, 1988], with the pooled standard deviation. Figure 3.7 shows the mean correctness and time for each technique all tasks combined as a function of N. Mean time is similar for the three techniques, but correctness drops as N increases for RLC (nearly 50% error) and HG, at a lower extent. We only report on significant effects that are summarized in Table 3.2, along with their effect size.

Table 3.2: Significant results for each factor by N and *task*. The best value for each line is in bold.

| N | Factor | Task | $F_{2,16}$ | p | Pairwise mean comparisons | Mean RLC | Mean HG | Mean IHG |
|---|---|---|---|---|---|---|---|---|
| 2 | *Time* | *Same* | 7.71 | * | HG ≪ RLC & HG ≪ IHG | 4.45s | **2.78s** | 3.80s |
| | | *Max* | 7.08 | * | RLC ≪ IHG & HG ≪ IHG | **2.77s** | 3.02s | 4.93s |
| | | *Disc* | 4.15 | * | RLC ≪ IHG & HG ≪ IHG | **3.30s** | 3.74s | 5.49s |
| 8 | *Time* | *Max* | 10.87 | ** | HG ⋘ IHG | 7.69s | **5.73s** | 11.40s |
| | | *Disc* | 5.45 | * | RLC ≪ IHG & HG ≪ IHG | **9.59s** | 10.18s | 14.45s |
| | *Correctness* | *Max* | 4.96 | * | RLC ≪ IHG | 0.833 | 0.972 | **1.0** |
| | | *Disc* | 9.45 | * | RLC < IHG | 0.805 | 0.944 | **1.0** |
| | *Error* | *Max* | 5.17 | * | IHG ≪ RLC & HG ≪ RLC | 7.43 | 0.73 | **0.0** |
| | | *Disc* | 6.15 | * | IHG ≪ RLC | 7.82 | 1.43 | **0.0** |
| 32 | *Time* | *Same* | 7.38 | * | IHG ⋘ RLC & HG ≪ RLC | 30.06s | 20.99s | **18.17s** |
| | *Correctness* | *Same* | 6.52 | * | RLC ⋘ IHG | 0.694 | 0.92 | **1.0** |
| | | *Max* | 10.20 | ** | RLC ⋘ IHG & RLC ≪ HG | 0.639 | 0.916 | **0.944** |
| | | *Disc* | 13.36 | ** | RLC ⋘ IHG & HG < IHG & RLC ≪ HG | 0.361 | 0.722 | **0.871** |
| | *Error* | *Max* | 9.61 | ** | IHG ⋘ RLC & HG ≪ RLC | 12.9 | 2.01 | **1.34** |
| | | *Disc* | 29.44 | *** | IHG ⋘ RLC & IHG ≪ HG & HG ≪ RLC | 24.15 | 9.01 | **3.23** |

* for $p \leqslant 0.05$, ** for $p \leqslant 0.001$, *** for $p \leqslant 0.0001$

We report Cohen-d's effect size [Cohen, 1988] computed using the pooled standard deviation:
$x < y$ for a *small effect* ($.2 < d < .3$), $x \ll y$ a *medium effect* ($.3 < d < .8$), $x \lll y$ a *large effect* ($.8 < d < \infty$).

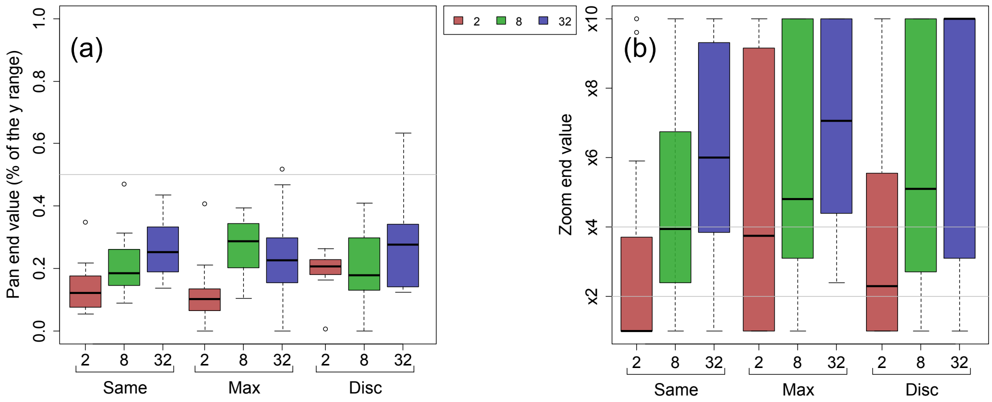



Figure 3.8: (a) Pan and (b) zoom values at the end of the trials by *task* and N for IHG. In (a), the grey horizontal line at 0.5 indicates the value of the baseline using HG (50% of the chart height). In (b), the grey horizontal lines at $z = 2$ and $z = 4$ are the recommended and the maximum values of b.

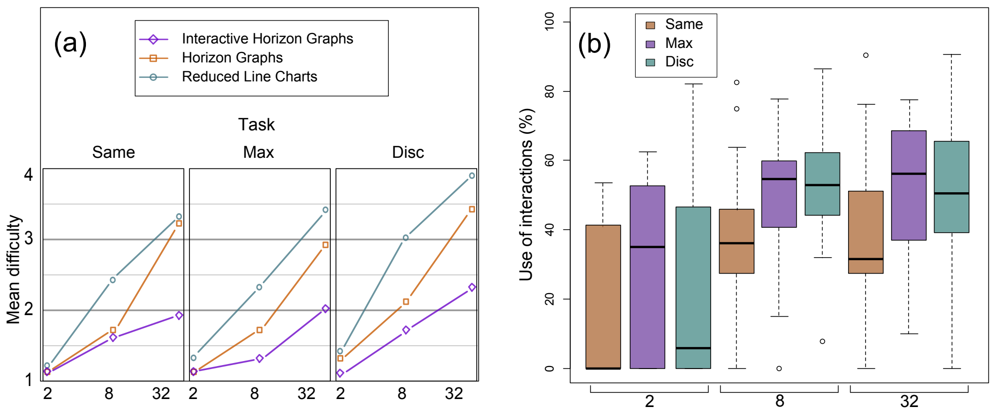



Figure 3.9: (a) Mean difficulty score for each *task* by N from participant's answers. (b) Pan and zoom use in percent of the trials total time for IHG.

### 3.4.1 *Use Of Pan And Zoom For Interactive Horizon Graphs*

Table 3.1 presents participants' use of pan and zoom for IHG. For *N=2*, half the participants did not use any interaction at all. For *N=8*, 71.7% used both types of interaction. For *N=32*, 86.7% used both. The harder the task, the more interaction was used. We also observed that for all N, few participants used only pan or only zoom—both seem useful to most participants. We recorded the values of the baseline and the zoom factor at the end of each trial for IHG (Figure 3.8) and the percentage of total time participants used pan and zoom (Figure 3.9(b)) using our kinematic logs. End values are important measures as they correspond to the number of bands and the value of the baseline the participants estimated to be the best for each trial.

### 3.4.2 *Questionnaire Results*

For each technique × task × N, we asked participants to give a score between 1 and 4 for *difficulty* (1: very easy, 2: easy, 3: difficult, 4: very difficult). Mean difficulty by task and N is reported Figure 3.9(a). All the 9 participants ranked the techniques in the same order regardless of the task and N: they ranked IHG first, HG second and RLC third.

## 3.5 SUMMARY AND DISCUSSION

The results confirmed our hypotheses that IHG were better than RLC and HG for large numbers of LSV time series.

### 3.5.1 *Influence of Number of Time Series*

In this subsection we detail the *statistically significant* differences between RLC, HG, and IHG for each N, and provide recommendations for the use of each technique.

**For *N=2*:** For *Same*, HG are faster than both RLC and IHG. This improvement is likely due to the fact that HG use colors that allow pre-attentive perception and recognition of key features. With IHG, participants lost time using the interactions, looking for recognizable shapes using pan and zoom.
For *Max* and *Disc*, both RLC and HG are faster than IHG: participants had been told that Correctness was more important than Time and we observed that they double-checked their answers using pan and zoom whenever they were in doubt. Figure 3.9(b) illustrates this observation—even for *N=2*, the use of pan and zoom represents up to 50% of the trials' time.

Because there is no difference in Correctness or Error for *N=2*, we recommend using HG for *N=8* or fewer. RLC can be used for elementary comparison and relation-seeking tasks such as *Max* and *Disc*. However, we do not recommend IHG for such small numbers of series because the interaction technique distracts users and does not bring any benefit.

**For *N=8*:** For both *Max* and *Disc*, HG are faster than IHG. The rationale is likely the same as for *N=2*—participants lost time using the interactions. Moreover, since the initial state of IHG was identical to RLC ($z = 1$, $y_b = y_m$), participants had to interact to obtain a visualization similar to HG, while for HG the default configuration was readily available. The remarkable distinction between *N=2* and *N=8* is that, in the latter, there are significant differences in Correctness and Error. For *Max*, IHG have higher Correctness than RLC because the zoom allows users to discern fine differences between charts. Since IHG and HG amplify the small variations, both techniques induce lower Error than RLC. For *Disc*, IHG have higher Correctness and lower Error than RLC for the same reasons.

In summary, IHG are 1.2 and 1.02 times more correct than RLC and HG for *Same* and 1.2 and 1.06 times more correct than RLC and HG for *Disc*. All participants completed the tasks with no error using IHG. We recommend using IHG or HG and avoiding RLC for medium numbers of time series when performing elementary comparison and relation-seeking tasks. The difference between HG and RLC was not highlighted in previous studies and is almost certainly due to the properties of our datasets.

**For *N=32***, both IHG and HG have higher Correctness and lower Error than RLC for all tasks except for *Same* where there is no difference in Correctness between HG and RLC. RLC are clearly limiting for large numbers of time series, regardless of the task. Interestingly, for *Disc*, IHG have higher Correctness and lower Error than HG. For this task—which is the hardest, involving visually browsing the charts vertically and horizontally—IHG exhibit better results than HG.

IHG are more correct than both RLC and HG for *Same* (1.4 and 1.1 times more), for *Max* (1.5 and 1.03 times more), and for *Disc* (2.4 and 1.2 times more). Not only are there significant differences between the techniques, but the effect size indicates that these differences are substantial. The Error measure also shows substantial differences: for *Max*, the Error for IHG is 9.6 times less than for RLC and 1.3 times less than for HG. For *Disc*, Error for IHG is 7.5 times less than RLC and 2.7 times less than for HG. This confirms that IHG leads to more correct answers and that, even when an answer is wrong, the Error is lesser than when using RLC and HG. For Time, there is no significant difference between IHG and HG regardless of the task. This is in contrast to the results for smaller N, where IHG were usually slower than the other techniques. Here, the overhead of interaction with the charts was less than that of visual search.

We strongly recommend using IHG for large numbers of time series and avoiding RLC. We also found that for large and medium numbers of time series, HG are more efficient than RLC, in contrast to previously published studies. Our work is the first to reveal these advantages of HG.

### 3.5.2 *Time vs. Accuracy*

The Time to perform *Max* and *Disc* is similar for all three techniques for *N=32* (Figure 3.7(b)) but the Correctness for RLC decreases severely between *N=8* and *N=32* (Figure 3.7(a)). Participants answered as quickly as in HG and IHG, but with very low Correctness. Participants' answers to our questionnaire explain this effect—for the RLC technique, their strategy was to quickly identify potential answers and to pick one randomly, without being sure of the answer. Clearly, regardless of how much time users take with RLC for *N=32*, they cannot perform *Max* and *Disc* correctly. We observed the same effect for HG , to a lower extent, but not for IHG. Figure 3.7(a) illustrates the scalability of each technique as a function of N, showing a clear advantage for IHG.

Figure 3.7(b) illustrates the Time to accomplish the task as a function of N. This shows a different trend than for Correctness—the Time for IHG and HG increases similarly with larger N, and the increase for RLC is much greater.

### 3.5.3 *Tasks*

As expected, Correctness decreases when N increases for all tasks. Furthermore, task difficulty can be clearly seen from the trends in Error: *Same* is the easiest task, followed by *Max*, with *Disc* being the hardest. Participants' questionnaire responses corroborate these results—they found *Disc* to be the hardest task and found that the *difficulty* dramatically increased with the number of time series (Figure 3.9(a)). These results are in agreement with Javed et al. [Javed et al., 2010]. However, our results do not show that HG are slower than RLC for *Max*, probably due to our use of LSV datasets.

### 3.5.4 *Hypothesis Control*

We confirm $\mathbf{H_1}$: *N=32* is the only value of N that showed clear differences between the three techniques. IHG have the highest Correctness and the lowest Error, followed by HG, while RLC was much worse. HG also have significantly better scores than RLC for both Correctness and Error. This difference had not been highlighted in previous studies and is explained by our use of LSV data—suggesting a need for multi-resolution techniques.

We reject $\mathbf{H_2}$: our results show that at least for task *Same*, IHG are significantly faster than RLC, but there is no significant difference with HG. This is due to the fact that, unlike HG, IHG require users to interact with the chart to obtain a useful configuration, which takes additional time.

We partially confirm $\mathbf{H_3}$: the Correctness for HG decreases when N increases and is lower than when using IHG. We did not find any significant difference between HG and IHG for *Max*, but IHG have substantially higher Correctness and less Error than HG for *Disc*. We were however surprised to see how robust HG are with respect to the number of time series; we did not expect such good results for this technique.

### 3.5.5 *Pan And Zoom*

**End-values:** Contrary to [Heer et al., 2009], the most useful zoom level can be well above 2. This can be seen in Figure 3.8(b), which shows $z$ at the end of each trial. We interpret the final value as being the most comfortable zoom level for answering the task.

For *Max* and *Disc* users' final zoom value is frequently the maximum zoom we allowed—10 bands. The recommended number of bands was rarely the one chosen for *N=8* and *N=32*. Our conclusion is that there is no default value for this parameter— the need for a higher or lower number of bands is related to the task, the dataset, and N. Conversely, the use of lower zoom values when completing *Same* can be explained by the strategy the participants adopted. Most participants modified the value of $y_b$ until a specific composition of color and shape appeared in the reference time series. Then they visually browsed all the time series to search this feature.

The *baseline* end value (Figure 3.8(a)) was rarely at the classic value of the baseline (50% of the chart height). This result is certainly due to the datasets, but confirms that if users have the possibility of modifying the baseline, they will choose a value which can be in a continuous range and will not limit their choice to a single value.

**Interactions:** The percentage of interaction time (Figure 3.9(b)) for *N=2* is low and does not linearly increase with N. Rather, it is about the same for both *N=8* and *N=32*—around 50% of the total time. This confirms that IHG are more useful for large numbers of time series but are distracting for *N=2*.

### 3.5.6 *Comparison With Previous Studies*

The differences between our study and the previous ones can be attributed to three factors: the use of interaction in IHG, the use of LSV datasets, and the use of the *Same* task instead of *Slope*. For *N=8*, contrary to previous studies [Javed et al., 2010], HG are significantly more efficient than RLC, likely because we used LSV datasets. Previous studies never tried *N=32* when all tasks become very difficult and interaction helps immensely. As for the choice of tasks, we have not compared IHG with the other techniques for *Slope* since this task was too hard to perform on LSV datasets, especially for RLC; the benefit of IHG on more uniform datasets remains to be studied.

Heer et al. recommended not to use too many bands [Heer et al., 2009] for value estimation tasks, not considered in our experiments. We are not sure value extraction would be accurate on LSV datasets, even with few bands.

### 3.5.7 *General Implications*

We used LSV datasets which are usually more challenging than the synthetic datasets used in previous studies, and also ecologically more valid. Our results show that more varied datasets should be used for future experiments to obtain more generalizable results.

We also believe that IHG can decrease the learning curve of HG because they start with the familiar RLC representation and, with continuous interactions using the pan and zoom, show novice users how HG are constructed. Our results highlight the fact that adding interaction to existing techniques can notably improve their performance as well as their usability.

### 3.5.8 *Limitations*

Our recommendations for design are valid under some conditions:

**Participants:** Our participants were students and researchers from HCI and Infovis and additional studies are required to evaluate IHG for novice users.

**N:** We constrained the number of time series to the height of a standard screen without having to scroll and more than 32 time series would require a larger screen.

**Datasets:** Our results are valid for LSV datasets, for which HG and IHG perform well. We have shown that IHG are efficient for at least one category of datasets, but a deeper range of datasets should be investigated.

**Tasks:** We did not consider value estimation tasks, since it requires users to perform a considerable amount of mental math using HG and IHG. However, alternative interaction techniques can be designed specifically to support value reading and extraction.

Finally, IHG suffer from the same limitations as Kronominer and marking menus for manipulating graphs that we discussed in Section 2.2. Indeed, the baseline panning and the values zooming interactions may conflict with other interactions for time series, such as the standard pan & zoom and the brushing selection that also involve dragging the mouse on the line chart canvas. Thus, it would require changing the interaction mode (P18) to avoid any interaction conflict.

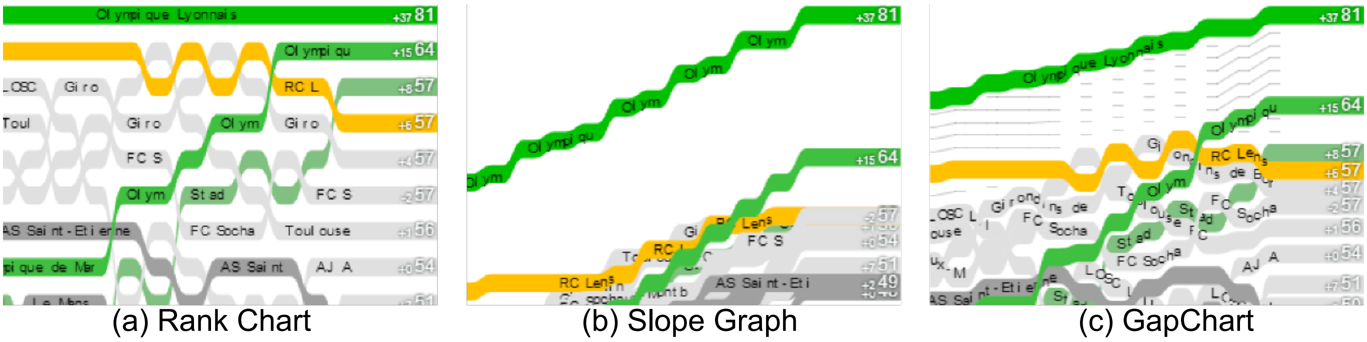



Figure 3.10: (a) Rank Chart prevents overlapping graphs, (b) Slope Graph shows the gaps between objects, and (c) GapChart as an intermediate visualization prevents overlapping graphs while showing the gaps between objects. The gray lines between objects in GapChart represent semantically meaningful gaps. Here, with soccer teams, each line represents a gap of three points, corresponding to a win in the championship.

## 3.6 CONCLUSION

We have presented *Interactive Horizon Graphs* (IHG), an efficient interactive technique for exploring multiple time series which unifies two split-space visualization techniques: *Reduced Line Charts* (RLC) and *Horizon Graphs* (HG). We have shown that IHG outperforms RLC and HG for several tasks in the most difficult conditions, thanks to interactive control of its two parameters: the baseline value and the zoom factor. Both relate to the number of bands traditionally used by HG. We have shown that IHG perform well with up to 32 time series, when previous work only tested up to 16. We also found that HG perform better than RLC for our datasets.

We conclude that systems visualizing time series using small multiples should provide our interaction techniques as a default. Our techniques generally improve performance on visual exploration tasks, except during the learning phase or for very small sets where interactions can be distracting.

Our contributions are: *(i)* the unification of RLC and HG by using interactive pan and zoom, *(ii)* a demonstration that IHG can scale up to 32 time series, and *(iii)* an evaluation using real LSV datasets rather than synthetic datasets with clear landmarks that help visual search tasks.

### 3.6.1 *Visualization Techniques Unification*

This work has shown that our simple interactions can unify two visualization techniques and substantially improve their efficiency. We hope it will be adopted to limit the proliferation of slightly different visualization techniques currently provided to explore multiple time series.

Several other visualization techniques can be unified using interaction. For example, we investigated the case of superimposed time series to visualize the ranking of items over time. Temporal rankings are prominent in a broad range of domains, such as academia (e.g., Shanghai University ranking), politics (e.g., election polls), demographics and economics (e.g., country ranks), and sport (e.g., Olympic games medals). In all cases, ranks at each time step are computed according to an underlying—often multidimensional—value and it is important to visualize both the rank of each element and the underlying value. Such temporal rankings are represented in newspaper and websites using a multidimensional table snapshot with updated values for each time step, making it impossible to analyze trends.

Overlaid line graphs are a common alternative to tabular data for visualizing the temporal evolution of rankings, and two families of line charts are used: Rank Charts and Slope Graphs. Both techniques map time to the x axis but they differ on the y axis mapping. Rank Charts map on the y axis the rank of each element for each time step and connect all ranks of an element with a line. Relying on pre-attentive capabilities, they make it easy to identify ranking changes (e.g., two intersecting line charts indicate that the two elements exchanged their rank). Although Rank Charts produce dense and overlapping-free visualizations, they do not show the gaps between elements, in terms of value difference. Slope Graphs are similar to Rank Charts, except that they map on the y axis the value of each element instead of the resulting rank. Slope Graphs emphasis the slope as the visual cue to convey important information (e.g., the slope between two time steps represents the values change). However, many empty areas appear and elements overlap when their value is equal, making reading their ranks difficult.

To understand the temporal evolution of ranking it is important to be aware of both the rank and the underlying value, as well as to visualize gaps between elements. For example, let's consider a soccer championship in the last days of the season. Using a Rank Chart (Figure 3.10(a)), the ranking of the teams is easy to read. However, the only way of estimating the magnitude difference between teams is to mentally subtract their values. Using a Slope Graph (Figure 3.10(b)) makes clear that the first team is far further the others, but it is impossible to determine which teams are ranked third to sixth, due to overlapping.

Our interaction technique impacts the $y$ axis of the graph and allows for a smooth transition between Rank Chart and Slope Graph. Interestingly, this interaction technique led us to discover—by accident—an intermediate technique that we call GapChart [13]. For a certain transformation of the $y$ axis, we can guarantee that there is no overlapping objects, while showing the gaps between objects over time. Moreover, an interesting property of the GapChart is that two objects with the exact same absolute value will be represented by two vertically contiguous objects, with no overlap, allowing for quickly discovering ties.

Using a GapChart for the above example (Figure 3.10(c)), ranks are easy to read even for the third to sixth teams. Moreover, the technique shows that these four teams had the same amount of points at the end of the championship (57), because there is no gap between them. The ranking was then computed according to an extra dimension (number of goals scored minus number of goals conceded). As for the Slope Graph, the GapChart also clearly shows the gap between the 1st and the 2nd teams of the championship but without any overlap.

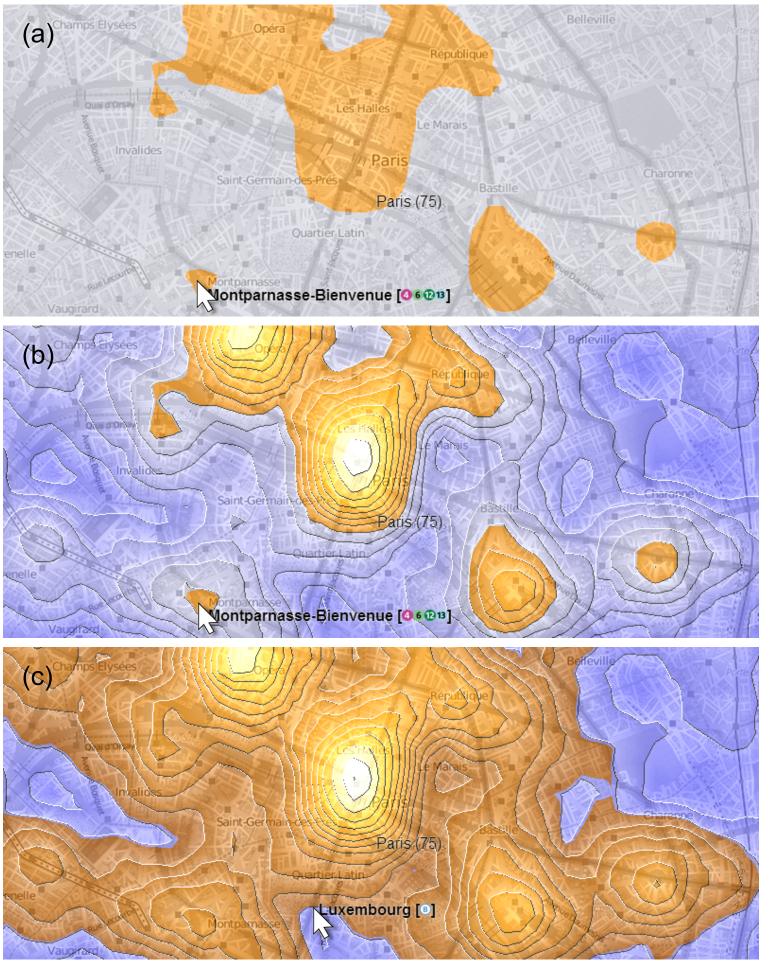



Figure 3.11: Interactive Horizon Maps. (a) One band and (b) Seven bands, both with the baseline set to the value of Montparnasse-Bienvenue; (c) Seven bands with the baseline set to the level of Luxembourg.

### 3.6.2 *Generalization*

Here we refer to the summary of principles, benefits and challenges summarized in Table 2.1. We proposed two interaction techniques for Horizon Graphs, that generalize to wide range of standard visualization techniques beyond time series visualizations. For example, barcharts and stacked area charts could benefit from these interactions.

The degree of integration of both baseline panning and values zooming interactions is 1/2 only (P13). Thus, we investigated contour maps as it is a spatial visualization technique. Figure 3.11 illustrates the two interactions for a contour map of a scalar field of Paris where values represent the distance to public transportation facilities. From (a) to (b), the zoom factor is changed from 1 to 7 using the mouse wheel, without changing the same baseline. The number of bands increases with the zoom factor, transforming the map into a contour map with an interactive number of contour lines.

The main difference with IHG is for the baseline panning interaction. Dragging the mouse in the 2D plane changes the level of the baseline, in the

Figure from Montparnasse-Bienvenue (b) to Luxembourg (c). Pressing the mouse button and dragging the mouse on the visualization sets the baseline value to the value of the scalar field under the cursor.

As a result, it provides an easy way of finding values and areas similar to a point of interest thanks to the contour lines and the color scales—more efficiently than using the common rainbow heatmap—by dragging the mouse cursor in the two-dimensional space. Thus, baseline panning interaction for 2D maps has a degree of integration of 2/2 and is a more congruent interaction than for IHG (P19).

### 3.6.3 *Implications for Interaction Design*

The two interactions we propose allow performing actions that are challenging for direct manipulation interfaces: it provides precise actions (C10) and identifying invisible object attributes without having to change the scale of the visualization (C6)—two challenges that Kronominer (Section 2.2.2.6) does not handle, for example. Standard panning and zooming could be an alternative, but for LSV datasets the interactions would be tedious and could disorient the user. Thanks to the wrapping metaphor, the context is always visible while small variations can be observed.

Beyond the evidence that simple interactions greatly enhance static visualization techniques, several lessons can be drawn from this chapter:

1. Interaction techniques can unify existing visualization techniques and limit the proliferation of slight variations.
2. It is important to try to design generic interactions that can be applied to a wide range of visual representations to limit the proliferation of interaction techniques.
3. Interaction techniques can trigger discoveries, as it is the case for the GapChart.

# 4 TEMPORAL NAVIGATION IN RANKING TABLES

This chapter is based on a previous publication [2]. Thus any use of "we" in this chapter refers to Charles Perin, Romain Vuillemot, and Jean-Daniel Fekete.

The previous chapter showed how interaction can *unify* two existing visualization techniques, such as Reduced line Charts and Horizon Graphs. In this chapter, we investigate the design of new interactions for a standard visualization technique: multidimensional tables that evolve according to time. We also explore how to *integrate* another visualization technique (a simple line chart) as an alternative view embedded into the table and accessible through interaction.

The questions we address in this chapter are:

- ▷ How to improve temporal navigation in ranking tables?
- ▷ What are the benefits of embedded direct manipulation interactions as compared to standard widgets?

This chapter introduces *À Table!*, an enhanced soccer ranking table providing temporal navigation by combining two novel interaction techniques. Drag-Cell is a direct manipulation technique to browse values over time by dragging them into the *Value* domain; Viz-Rank uses a line chart to show the values of cells over time, in the *Time* domain. Both techniques follow Bederson [Bederson, 2004] principles to preserve users' flow, to focus on tasks and prevent interruptions.

Every day, millions of soccer enthusiasts study web sites, newspapers, or other media to learn about their favorite team's current ranking and how it compares historically. One way to present this information is to use ranking tables that show results for a given championship, at a given day. Ranking tables order soccer teams represented as rows, according to values of columns containing attributes. Column types include, for example: the total points per season, number of wins, draws, and lost games. Because they represent a snapshot of a championship at a time $t$, tables are regularly updated with new results. Tables are updated once a round of games is over, and can become quite large. Assuming a championship with 20 teams, a table will be updated 38 times since each team plays against each other twice. Such updates usually change the rows order, which makes the tracking of a specified team over time difficult.

We conducted an empirical study of 51 articles (weekly and monthly summaries of soccer championships) from major soccer websites, and collected 44 pseudo-interactive soccer tables to investigate how they support temporal tasks. Our first observation was that ranking tables are not designed for comparison, despite the need for analysts and the existence of guidelines [Rao and Card, 1994; Tufte, 1986]. Furthermore, 77.3% of tables do not implement column sorting, and 73% of them do not provide any aid for temporal navigation although it may be useful for interactive exploration [Gratzl et al., 2013; Rao and Card, 1994]. Finally, we did not observe any use of visualization, however it provides compact representation for multidimensional data [Gratzl et al., 2013; Rao and Card, 1994] and displays trends over long time periods with line charts [Playfair et al., 2005; Shi et al., 2012].

We conducted an on-line evaluation to assess how standard interactions with ranking tables (sorting and changing time with a slider), along with the two novel techniques, Drag-Cell and Viz-Rank, can be effective for temporal tasks. Our evaluation shows that the novel techniques efficiently support temporal tasks, which are currently not supported by ranking tables. We discuss the design implications of our improvements and give some scenarios to apply *À Table!* to other domains of application.

Rank according to number of points

Number of points

Dimensions to compute the number of points

| | | Team | Pts | W | D | L | Ho | Aw | GF | GA | GD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | — | Paris-SG | **83** | 25 | 8 | 5 | 43 | 40 | 69 | 23 | 46 |
| 2 | — | Marseille | **71** | 21 | 8 | 9 | 40 | 31 | 42 | 36 | 6 |
| 3 | — | Lyon | **67** | 19 | 10 | 9 | 37 | 30 | 61 | 38 | 23 |
| 4 | ▲ | Nice | **64** | 18 | 10 | 10 | 38 | 26 | 57 | 46 | 11 |
| 5 | ▼ | Saint-Etienne | **63** | 16 | 15 | 7 | 37 | 26 | 60 | 32 | 28 |
| 6 | ▼ | Lille | **62** | 16 | 14 | 8 | 34 | 28 | 59 | 40 | 19 |
| 7 | ▲ | Bordeaux | **55** | 13 | 16 | 9 | 32 | 23 | 40 | 34 | 6 |
| 8 | ▼ | Lorient | **53** | 14 | 11 | 13 | 36 | 17 | 57 | 58 | -1 |
| 9 | ▼ | Montpellier | **52** | 15 | 7 | 16 | 38 | 14 | 54 | 51 | 3 |
| 10 | — | Toulouse | **51** | 13 | 12 | 13 | 28 | 23 | 49 | 47 | 2 |
| 11 | ▲ | Valenciennes | **48** | 12 | 12 | 14 | 33 | 15 | 49 | 53 | -4 |
| 12 | — | Bastia | **47** | 13 | 8 | 17 | 30 | 17 | 50 | 66 | -16 |
| 13 | ▼ | Rennes | **46** | 13 | 7 | 18 | 23 | 23 | 48 | 59 | -11 |
| 14 | — | Reims | **43** | 10 | 13 | 15 | 31 | 12 | 33 | 42 | -9 |
| 15 | — | AC Ajaccio | **42** | 9 | 15 | 14 | 26 | 16 | 39 | 51 | -12 |
| 16 | ▲ | Sochaux | **41** | 10 | 11 | 17 | 23 | 18 | 41 | 57 | -16 |
| 17 | ▼ | Evian-TG | **40** | 10 | 10 | 18 | 25 | 15 | 46 | 53 | -7 |
| 18 | ▲ | Nancy | **38** | 9 | 11 | 18 | 20 | 18 | 38 | 58 | -20 |
| 19 | ▼ | Troyes | **37** | 8 | 13 | 17 | 26 | 11 | 43 | 61 | -18 |
| 20 | — | Brest | **29** | 8 | 5 | 25 | 18 | 11 | 32 | 62 | -30 |

Last team evolution    Team logo

Figure 4.1: Illustrative example of a "complete" table at time t, featuring the most frequent designs and embedded visualizations. Pts: number of points; W, D, L: number of Wins, Draws, Losses; Ho, Aw: number of points at Home and Away; GF, GA, GD: Goals For (scored), Against (conceded), Difference (GF-GA).

The contributions of this chapter are as follows:

1. A Domain description and task analysis for temporal navigation techniques.
2. Two new techniques for temporal navigation in ranking tables, one in the value domain (Drag-Cell) and one in the time domain (Viz-Rank).
3. A crowdsourced evaluation of the two techniques along with standard navigation techniques; we then discuss which technique is well suited to which task according to the results and present the main takeaways from the experiment.

Such results provide a framework for further experiments with temporal tasks, and the design of novel navigation techniques in both the *Value* domain and the *Time* domain. Numerous application areas where ranking tables are important would benefit from these results.

## 4.1 BACKGROUND AND RELATED WORK

Soccer ranking tables (Figure 4.1) are standard ranking tables, i. e. tabular data vertically ordered according to values in *one* column. Ranking tables have been extensively explored using summarization, interaction and visualization techniques. As far as we know, temporal exploration in tables has never been investigated in a comprehensive way.

A rank is an ordering technique taking as input a set of items $S$ (e. g., teams) and provides a permutation of these items according to one dimension (e. g., points, goals scored). A rank is a function $\textsc{rank} : S \to 0, ..., |S| - 1$, that generates up to $|S|!$ permutations, i. e. the number of ways the items in the set can be uniquely ordered [Kreher and Stinson, 1999]. Ranks apply to dimensions (columns) $D$, $|D| > 0$. Finally, a rank can be applied to *temporal* values, where $t \in T$, with $T$ a set of discrete events (e. g., games in a championship). $S_{d_i,t_j}$ is the ranking table, which is a *snapshot* of a championship, ranked according to a dimension $d_i$ at a time $t_j$.

A soccer ranking table usually has the following properties: $|S| = 20$ teams; $|D| = 10$ dimensions; $|T| = (|S| - 1) \times 2$, i. e. $|T| = 38$ games; $0 \leqslant t \leqslant 38$. An important property of permutations in soccer championships is that the higher $t$ is, the less the teams' permutation amplitude is important because teams tend to have high points difference.

Ranking tables display tabular data and their visual design normally obey to guidelines. Tufte [Tufte, 1986] suggests that minor visual improvements, such as vertical alignment of characters and row coloring with zebra patterns, enable comparison tasks and reduce errors. Interacting with the rank *order* help users perform some tasks (e. g., finding maximum or minimum values) faster, without scanning all the rows. Many software packages, such as spreadsheets, implement ranking interactions. They also provide formulas to summarize rows for more compact representations with *Pivot tables* [Wikipedia, 2013] by computing counts or totals. The same principles can be applied to columns with statistics such as averages of multiple columns or min/max values to provide other ranking mechanisms [Rao and Card, 1994].

Summarization of tables is an important challenge because rankings are often longer than the screen. Visualization techniques can provide compact representation of cells. TableLens [Rao and Card, 1994] represents tables using Focus+Context principles, collapsing rows to their minimum size or up to a pixel. This enables representing large tables on one screen as long as the number of rows is less or equal than the number of pixels. However, one row is always expanded as a focus and the content of the cells is visible. LineUp [Gratzl et al., 2013] also uses nested compact visualizations in tables, but for headers showing the distribution of values contained in the rows beneath.

Ranking also plays an important role in discovery, particularly when tables are represented as a grid layout where visual variables such as color encode values. It results in heatmap-like representations [Sopan et al., 2010], and ranking becomes the primary interaction for finding patterns [Seo and

Shneiderman, 2005]. Such tables may become very large and require space transformation techniques, such as zooming or space folding techniques [Elmqvist et al., 2008b] to bring rows and columns back together for comparison.

Visualization techniques to display ranks are mainly inspired by line charts [Playfair et al., 2005], and are also called *bump charts* because of the visual effect that permutations provide. Slope Graph [Tufte, 1986] plots items' *values* on two vertical axis—one for each time $t_1$ and $t_2$—and connects similar items from one axis to another with a line. The slope of the line is a visual cue that conveys the importance of the items' value change. However, some empty space may appear within the chart and elements may overlap. Several recent attempts produced compact charts reducing the amount of space and overlapping, while handling scalability. RankExplorer [Shi et al., 2012] shows items ranking as a flow, with glyphs wherever two items swap their ranks, or when an item has a particular increase in rank. Generally, visualization techniques are low-dimensional projections of data, that do not keep the original multidimensional flavor and interactivity of tables. They may also break users' flow [Bederson, 2004] because tabular layouts are too different from line charts, which require a cognitive overhead for users to connect both.

In summary, existing table improvements tackle the challenge of better sorting, summarizing, and visualizing tables. These techniques are usually not trivial for a non-expert, which may explain why we did not find any in our empirical study. Another reason might be that techniques are not specifically tailored for temporal navigation, as time is usually a tricky dimension that requires specific attention.

## 4.2 EMPIRICAL STUDY OF SOCCER RANKING TABLES

We conducted an empirical study to better understand current soccer ranking tables design practices. Over the 44 tables we collected, we observed that temporal navigation is rarely supported, unless the table uses non-dynamic drop-down lists. We also investigated the use of time in soccer newspapers by collecting 51 articles. We realized that they constantly refer to temporal trends or team performances during a time period, that current table designs do not support well.

### 4.2.1 *Time in Soccer Articles*

We collected and analyzed 51 soccer articles, mainly from two highly influential and visited websites: L'équipe[1] (70%) weekly summaries for the entire 2012/2013 Ligue 1 Championship in France; and premierleague.com[2] (20%) monthly summaries of the 2012/2013 Premier League Championship in the UK. This was completed with articles from Yahoo sports, Wikipedia and other soccer websites (10%). The articles we collected outlined main events of the week or month, and illustrated them with statistics.

Out of the articles, we extracted up to 33 different tasks the journalists had to perform to write these summaries. Then, we grouped similar tasks into *generic* categories. For example, both tasks *"What is the number of points of Paris SG at week 17?"* and *"How many goals has Marseille scored until week 5?"* are subtasks of the generic category *"What is the `VALUE` of `TEAM` at `t`?"*, where `TEAM` and `t` are the two known variables and `VALUE` the unknown one. This grouping resulted in 18 generic task categories.

We observed that all the tasks categories are temporal. Indeed, even the most basic task, such as *"What is the `VALUE` of `TEAM`?"* implies knowing when to look at the value. When t is not explicitly written, it is implicitly the latest week of the championship. We also observed that the most important column for ranking tables is the number of points of a team, determining its rank in the championship. However, the other dimensions also occur, to improve the summary details and analysis. For example, analysts often report: the number of scored goals or wins for the last n games of a team; the team's performances at home; or a series of consecutive games with the same result (e. g., 5 losses in a raw).

To structure our analysis, we mapped the categories to the task taxonomy for temporal data by Andrienko and Andrienko [Andrienko and Andrienko, 2005]. Figure 4.2 illustrates the mapping and provides examples. They classify the tasks into two categories. 1) *Elementary tasks* are *local* tasks where the object of interest is a given value or a given time. For example, such tasks are value estimation (e. g., finding extrema [Amar et al., 2005]), comparison of values at different times and time retrieval according to a given value. 2) *Synoptic tasks* are *global* tasks involving the user to take into account a set of

1 http://www.lequipe.fr
2 http://www.premierleague.com/

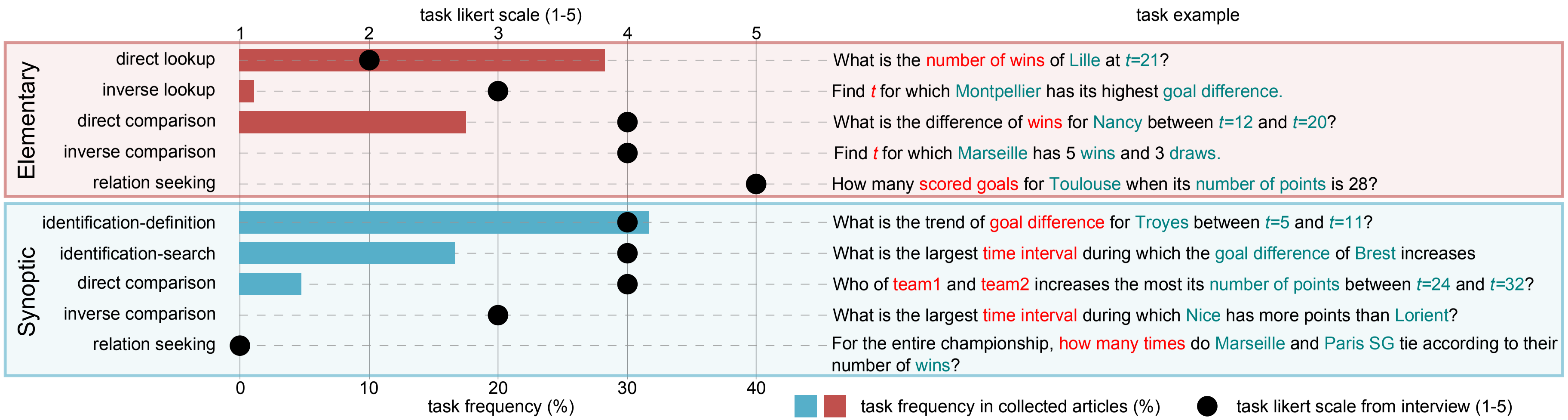



Figure 4.2: Extracted task categories mapped to Andrienko and Andrienko task taxonomy for temporal data, in percent of the collected articles. Black dots indicate how a soccer expert analyst estimates each category's relevance on a Likert scale from 1 (not relevant at all) to 5 (highly relevant). On the right are exemplary tasks for the categories. Red words are the unknown variable, blue words are the known variables.

values or a time interval. For example, such tasks are trend estimation, trend comparison, and finding time intervals with a given trend. Both categories contain *direct* and *inverse* tasks. Direct tasks consist of estimating a value or a trend according to a given time or time interval. Inverse tasks consist of estimating the time or time interval for which a dimension has a given value or trend. Finally, *relation-seeking* tasks consist of finding relations between dimensions, values and time. For example, such a task consists of comparing the value of a dimension for one team with the value of another dimension of another team, at different times.

We also conducted an interview with a soccer analyst in charge of an influential French web site[3] and asked him to evaluate the fitness of each task category for soccer (Figure 4.2, Likert scale values). One interesting result is that the expert considers some *categories* to be important, despite not found in articles (e. g., elementary relation-seeking tasks), suggesting that those tasks are probably unconsciously in the head of our expert but that he can not perform them with his current tools.

The expert and the task analysis together legitimate the tasks that we classified in the taxonomy. The tasks our expert mentioned are, according to him, difficult to perform. In particular because they involve ranks that change over time, which are difficult to track with series of static tables. This suggests that journalists have important questions that they are not easily able to answer. Consequently, we focus on supporting tasks and analysis questions that occur in this particular domain (i.e. temporal navigation in ranking tables) in the rest of the chapter.

### 4.2.2 *Soccer Ranking Tables Visual and Interactive Design*

We collected a series of 44 ranking tables for soccer, from UK (39%), France (35%), Spain (11%), Argentina (9%), Brazil and US (3% each). We extracted and ranked the visual features of the collected tables, such as colors and decorations.

An example of an important feature in soccer tables is the background color for the top-k ranks, such as the top-3 ranks that qualify for the Champion's League. Or the bottom-k that indicates which team will downgrade to a minor league. Such an apparently small visual aid is actually very useful, to immediately know which team will earn additional revenues with Champion's Leagues games, or will lose revenues by playing in a less popular league the next year.

For the top-3 rank, we found that only 50% of the tables highlight the corresponding areas, and their design is inconsistent (e. g., icons, colored rows, bold/dashed separators, colored text, and gray scale). Tables also make use of Zebra (50%) and embedded visualizations: 27.3% show team logos, 29.5% add an icon encoding the latest ranking evolution, and 13.6% represent the latest results using colored circles. As a reference, Figure 4.1 shows a *full* table, with the most frequent features.

---
3 http://www.cahiersdufootball.net/

One of the main takeaways from our findings is that interaction is rarely available. At best, the table provides the most common column sorting (for 22.7% of analyzed ranking tables). At times, the table provides widgets to interact with the temporal dimension: dropdown lists (11%), range selection (6.8%), next/previous arrows (4.5%), discrete slider (2.3%). However, it usually triggers a page reload, making the tracking of changes between the two tables difficult.

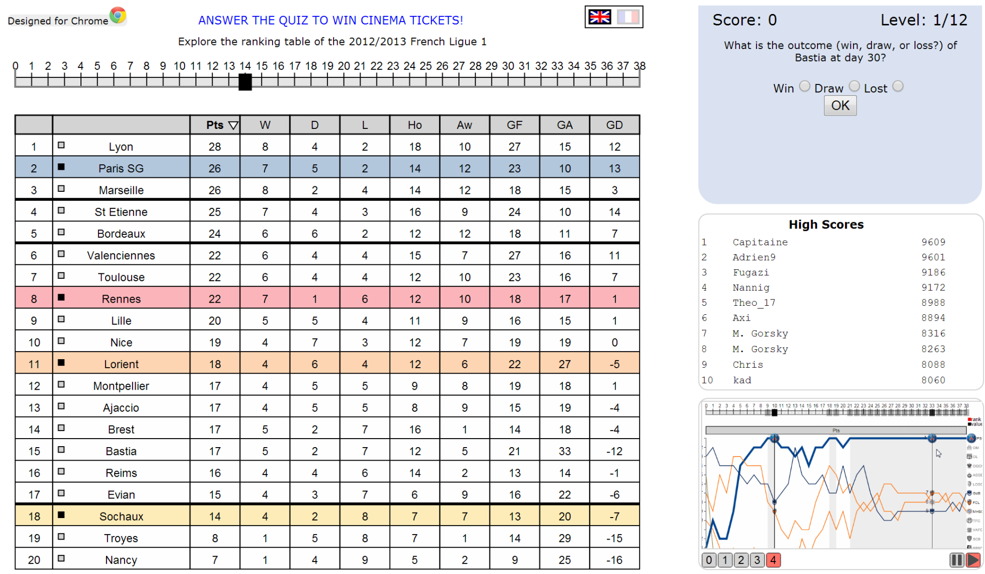
We also observed an interesting type of ranking table mapping rows on an absolute scale (linear scale of values, in this case points) rather than the relative scale (linear scale of ranks). It is referred as *Absolute scale rank*. Similarly as for Slope Graph [Tufte, 1986], it gives a better idea of the team's distribution at a given time. However, it requires more space as the count of rows for such a table is not the number of teams, but the difference of points between the first and the last team.

### 4.2.3 *Time Navigation*

Only 27% (12/44) of the tables implement temporal navigation while both soccer ranking tables and analyst tasks heavily rely on time. When available, changing $t$ is usually cumbersome, with standard widgets—certainly because implemented in every GUI—such as:

- Drop-down lists to directly jump to a specific time. They cannot be used to dynamically browse a time interval but are very efficient to select a particular time.
- Arrow widgets to jump to previous or next tables, with some shortcuts to the beginning or the end of the championship. They are efficient to navigate sequentially, step by step.
- Range slider for 6.8% (3/44), to select the data in $[t_i, t_j]$ and compute the cell values between these two times.
- slider for 2.3% (one table) to support temporal navigation, similarly to a seeking bar to scrub videos.

The results from our study highlight the lack of temporal navigation mechanisms, and only one table featured a slider which would be recommended as an efficient way to explore such a quantitative data space [Shneiderman, 1994]. We are not able to give any explanation. From a technical perspective, standard widgets are implemented in Hypertext Markup Language (HTML) and the data to compute ranking tables are relatively small and freely available. From a user perspective, we showed a dynamic slider to our soccer expert and he found it of great value, wondering himself why he does not provide one on his website.




| | | Pts ▽ | W | D | L | Ho | Aw | GF | GA | GD |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | □ Lyon | 28 | 8 | 4 | 2 | 18 | 10 | 27 | 15 | 12 |
| 2 | ■ Paris SG | 26 | 7 | 5 | 2 | 14 | 12 | 23 | 10 | 13 |
| 3 | □ Marseille | 26 | 8 | 2 | 4 | 14 | 12 | 18 | 15 | 3 |
| 4 | □ St Etienne | 25 | 7 | 4 | 3 | 16 | 9 | 24 | 10 | 14 |
| 5 | □ Bordeaux | 24 | 6 | 6 | 2 | 12 | 12 | 18 | 11 | 7 |
| 6 | □ Valenciennes | 22 | 6 | 4 | 4 | 15 | 7 | 27 | 16 | 11 |
| 7 | □ Toulouse | 22 | 6 | 4 | 4 | 12 | 10 | 23 | 16 | 7 |
| 8 | ■ Rennes | 22 | 7 | 1 | 6 | 12 | 10 | 18 | 17 | 1 |
| 9 | □ Lille | 20 | 5 | 5 | 4 | 11 | 9 | 16 | 15 | 1 |
| 10 | □ Nice | 19 | 4 | 7 | 3 | 12 | 7 | 19 | 19 | 0 |
| 11 | ■ Lorient | 18 | 4 | 6 | 4 | 12 | 6 | 22 | 27 | -5 |
| 12 | □ Montpellier | 17 | 4 | 5 | 5 | 9 | 8 | 19 | 18 | 1 |
| 13 | □ Ajaccio | 17 | 4 | 5 | 5 | 8 | 9 | 15 | 19 | -4 |
| 14 | □ Brest | 17 | 5 | 2 | 7 | 16 | 1 | 14 | 18 | -4 |
| 15 | □ Bastia | 17 | 5 | 2 | 7 | 12 | 5 | 21 | 33 | -12 |
| 16 | □ Reims | 16 | 4 | 4 | 6 | 14 | 2 | 13 | 14 | -1 |
| 17 | □ Evian | 15 | 4 | 3 | 7 | 6 | 9 | 16 | 22 | -6 |
| 18 | ■ Sochaux | 14 | 4 | 2 | 8 | 7 | 7 | 13 | 20 | -7 |
| 19 | □ Troyes | 8 | 1 | 5 | 8 | 7 | 1 | 14 | 29 | -15 |
| 20 | □ Nancy | 7 | 1 | 4 | 9 | 5 | 2 | 9 | 25 | -16 |

Figure 4.3: *À Table!* integration into a web page for on-line evaluation. The interactive table along with the temporal slider is on the left; the quiz, the high scores and navigable animated GIFS as explanations on the right.

## 4.3 IMPROVING TEMPORAL NAVIGATION

*À Table!* is an advanced ranking table that includes current ranking table practices (Figure 4.3), with two novel techniques for temporal navigation: DRAG-CELL is based on direct manipulation of values to browse ranks; VIZ-RANK uses a transient visualization of team ranks (line chart) to explore a championship. Those techniques aim at improving the support of temporal tasks we introduced in Figure 4.2.

### 4.3.1 *Features from Current Ranking Tables*

*À Table!* is a ranking table, along with the following features:

- *Temporal Slider* to give both an overview of the championship, and to provide a continuous temporal navigation. The slider is also synchronized with the two new interaction techniques we describe below to convey visual feedback.
- *Multiple teams selection* by clicking on team names to select and highlight their row, to facilitate their tracking over time.
- *Column sorting* by clicking on the table's headers to apply the `rank` function over a specific dimension.

### 4.3.2 *Design Philosophy for the Novel Techniques*

We designed DRAG-CELL and VIZ-RANK with the challenge of preserving users' flow. We followed Bederson [Bederson, 2004] principles by supporting both novice and experts with the same display. While tabular display remains the default view, novel techniques are activated with specific mouse

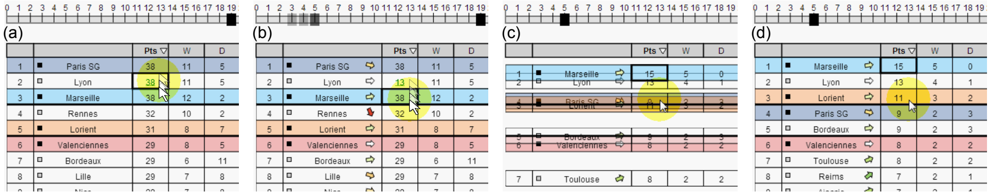


Figure 4.4: DRAG-CELL: (a) dragging up and down the value of a cell (in this case, Points of Team Lyon) makes this value change (b) for the set of values it can have, and colored arrows indicate how teams would behave if the drag was released (green to red arrows). (c-d) Releasing the drag animates the rows to their new position.

interactions and new representations are animated to prevent users' interruptions. User is always in control with incremental actions and a visual feedback to constantly show what he is doing.

Because the design space for novel techniques is important, we decided to focus on two techniques which support specific temporal tasks that ranking tables or standard widgets do not currently support: DRAG-CELL lets the user interact in the *value domain* and VIZ-RANK in the *time domain*.

### 4.3.3 *Novel Technique 1: Direct Manipulation of Values*

From the task analysis, we observed that the expert estimated as important several inverse tasks which were not frequent in journalist articles. Inverse tasks are challenging to perform using standard tables and it may explain why journalists avoid these tasks. We designed DRAG-CELL to make easier to perform these difficult tasks by interacting with cell values directly (e. g., the number of points of a team): the user manipulates the *value domain* instead of the *time domain*, unlike standard navigation techniques such as a temporal slider. Typically, DRAG-CELL allows users to quickly find if and when a value was reached. For example, a frequent task consists of finding the time when a team reaches 42 points; this specific value being the theoretical minimum number of points a team needs to be safe from the downgrade area.

DRAG-CELL follows the principle of direct manipulation [Shneiderman, 1983], with teams as objects of interest and is inspired by Tangle [Victor, 2013b]. The visual instance of teams are rows, and each cell can be dragged to explore the values of the cell itself over $t$ (Figure 4.4). Releasing the drag changes the value of $t$ for the table to the local $t_c$ of the cell. The user's flow is as follows:

1. **Start:** DRAG-CELL is activated by a mouse *drag* on a cell.
2. Then by dragging the mouse up and down, the cell value changes for all possible values for this cell.
3. Arrows indicate the preview of each team's behavior over time if the drag was released: the arrows range from pointing down and red (e. g.,

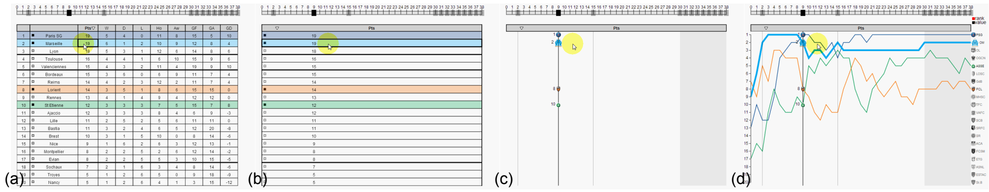


Figure 4.5: Table to VIZ-RANK: (a) clicking on a cell of a team for a dimension makes (b) the dimension's column grow horizontally, stretching the other columns until they disappear; then (c-d) the cells fade out and the time series of the cell's value for the team fades in. Line charts with a thinner stroke shows previously selected teams' line charts.

the team is going down a lot) to heading up and green (e. g., the team is going up a lot).

4. Additionally, the temporal slider displays a visual feedback as a preview of the current $t_c$.
5. **End:** Once the drag is released, $t_c$ is applied to the table. Rows permute with animation to their new rank position.

### 4.3.4 *Novel Technique 2: Line Charts as Transient Objects*

VIZ-RANK displays a temporary or *transient* line chart when a user clicks on a cell (Figure 4.6). This line chart displays the teams' ranks over the whole championship. The user can click on one point on the chart and t is set according to the corresponding value. The user's flow is as follows:

1. **Start:** VIZ-RANK is activated by a mouse *click* on a cell.
2. Then, an animated staged transition [Heer and Robertson, 2007] transforms the table into a time-line format by successive widening of columns and rows (Figure 4.5).
3. A line chart is displayed with the X-axis encoding the whole championship time range and the Y-axis the dimension of the cell the user clicked on. Each team line is encoded with Semantically-Resonant Colors [Lin et al., 2013] that matches teams' logo (which is also displayed at the end of the lines).
4. The line chart displays by default the ranks, but the Y-axis can also encode the absolute values (Figure 4.6).
5. A visual feedback displays a vertical bar at the current mouse position to enable comparison. The team associated to the cell the user clicked on is highlighted.
6. **End:** When clicking on the chart, it is animated back to its table form with t corresponding to the click x coordinate. The staged transitions are similar to the ones from the table to the line chart, but in reverse order.

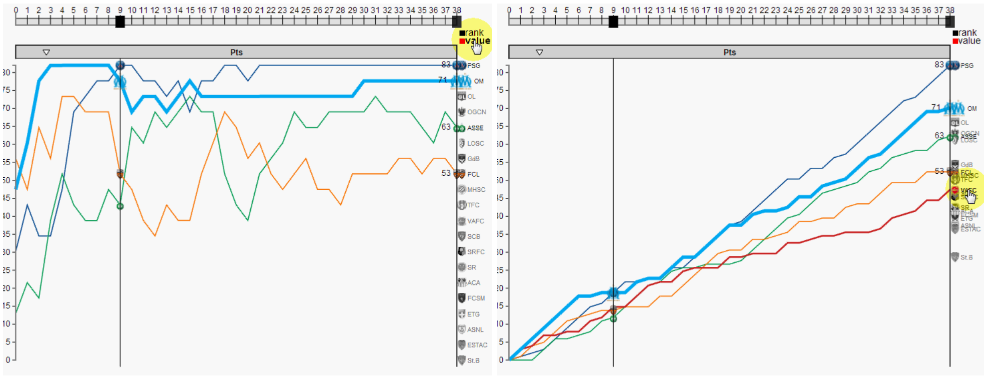


Figure 4.6: The two available line chart types: Rank (left) or Value (right). A click on a team's name or logo on the right selects the team.

## 4.4 Crowdsourced Evaluation

We conducted an evaluation to assess how standard interactions, along with Drag-Cell and Viz-Rank, can be effective for temporal tasks. We also expect to identify which task categories discriminate the techniques.

### 4.4.1 *Methodology*

We released *À Table!* on the web during a 3-week period, for a crowdsourced evaluation. *Crowdsourcing* is the use of an unknown public to perform simple tasks [Quinn and Bederson, 2011]. Participants (workers) are usually recruited through online markets (e. g., Amazon's Mechanical Turk) and paid to obtain results of quality. We crowdsourced the evaluation because the target user is specific (soccer enthusiast with interest in statistics) and to keep unaltered their environment (they usually browse specialized websites) without paying them. We also picked up this evaluation method because existing tables do not support dynamic temporal navigation and have very inconsistent designs, thus a traditional lab experiment with a baseline comparison would not be appropriate.

We evaluated participants' performance using a quiz (Figure 4.3, top right). We designed the quiz's questions based on the extracted tasks for soccer analysis (similar as the examples in Figure 4.2). We also displayed the quiz high scores as an incentive to engage visitors.

### 4.4.2 *Implementation*

*À Table!* is implemented using JavaScript and *D3* [Bostock et al., 2011] to run in any modern web browser. The web page is available in English and French at `https://github.com/charlesperin/atable`. We implemented the table and the temporal slider on the left; the quiz and the high scores on the right (Figure 4.3). We also added a tutorial as instructions for the novel techniques on the bottom right, using a slide-show of animated GIFs. A help button at the bottom opened a help page on how to use the table. The experimental setting is available at `http://charles.perin.free.fr/atable`. We collected feedback from participants with an on-line questionnaire, reached either by clicking on a feedback button or once the quiz is completed. We used the data from the French Ligue 1 2012/2013 championship.

### 4.4.3 *Hypotheses*

Our hypotheses for the experiment are as follows:

- **H1**: side tutorials will help users to activate and learn the novel techniques, even without visual cues or affordances on the table.
- **H2**: the temporal slider will be faster for elementary direct lookup/comparison because these tasks only require changing t and browsing the table.
- **H3**: Drag-Cell will be the fastest for elementary inverse lookup/comparison tasks, tasks it was designed for.
- **H4**: Viz-Rank will be errorless for synoptic tasks because it expresses best the temporal evolution of values over the whole time span.
- **H5**: Viz-Rank will be slower for all tasks because the mental representation of the teams for users changes.

### 4.4.4 *Tasks*

Participants performed tasks issued from each category (Figure 4.2) using our previously set of 33 tasks. We equally distributed elementary and synoptic tasks. Task variables were randomly generated with some constraints to avoid trivial tasks where the participant's knowledge of the data would be enough to answer. Participants had to answer 36 questions correctly to complete the quiz.

### 4.4.5 *Participants*

We recruited participants by advertising the web page URL using social networks, mailing lists and soccer forums, to select appropriate participants interested in soccer and statistics.

*Incentive Impact:* after two weeks, among the 141 visitors who resulted in 242 tasks performed, only **one** fully completed the quiz and answered the questionnaire. We observed the following recurring pattern: most visitors interacted with the table, answered a few questions and then dropped out without finishing the quiz. To encourage visitors to answer more questions and eventually complete the quiz, as an incentive we advertised free cinema tickets to participants with the highest scores. This immediately resulted in a stronger engagement from the participants.

*Demography:* we extracted the following informations from Google Analytics: most of the visitors were from France (60.5%), then United States (17.4%), United Kingdom (4.9%) and Canada (3%). 53.1% used French and 46.9% English.

### 4.4.6 *Participation Logs*

We logged all participants' interactions, such as: column sorting, team highlighting, slider navigation, DRAG-CELL and VIZ-RANK interactions. We recorded the following values for each quiz answer: quiz session, question number, id, and category; time to perform the task; participant's answer; correct answer; associated interactions ids.

### 4.4.7 *Results*

Over a 3-week period, we registered 1292 visits and 648 visitors performed at least one task. We discarded the data for participants who performed less than 10 tasks and the 239 answers longer than 120 seconds. 143 performed more than 10 tasks (*G1*), 62 participants completed the quiz (*G2*) and 34 who completed the quiz also filled the questionnaire. This results in 6693 tasks performed and 185 636 interaction logs.

*Quantitative results:* participants used VIZ-RANK for 31% of their answers, DRAG-CELL for 9%, and exclusively the other features for 60%. We performed the quantitative analysis using *G1* results—the larger dataset—because we did not find any significant difference between *G1* and *G2*. Figure 4.7 shows the time and error for all task categories according to performed interactions. We grouped the answers from participants using VIZ-RANK and other interactions into *VR*, DRAG-CELL and other interactions into *DC*, and other interactions (exclusively) into *O*. When both DRAG-CELL and VIZ-RANK were triggered, we counted the last interaction as the one that led to the answer. We performed Anovas when the data had a normal distribution (we applied a log transform to the measures of time) and when the analysis of variance allowed it (Bartlett's K-squared). We used a Welsh two-sample t-test (unequal sample size and variance) for pairwise means comparison. We report significant results only in Table 4.1.

*Interactions:* Figure 4.8 shows which interactions were performed, for correct and incorrect answers: all interactions have a higher percentage of use for correct answers than for incorrect ones. The most frequent interaction is changing the value of t (for more than 80% of correct answers) and was triggered most using the temporal slider, then VIZ-RANK and DRAG-CELL. Participants used the team selection for 70% of the tasks, and column sorting for a third.

*Qualitative Results:* Figure 4.9 shows the participant's scores on Likert scales for their background, questions on the interface and more specific questions about *À Table!* features. The main observation is that participants' satisfaction with existing ranking tables is low: 91% (31/34) of the participants would like to have access to such an interactive table on their favorite soccer website. Indeed, they estimate it offers better ways to analyze soccer championships (74% of the participants discovered unknown information during the quiz). Interestingly, they scored the team selection, the column sorting and the temporal slider as the most useful features. However, none complained that the novel techniques were distracting, or that they were disturbing obstacles to perform the tasks.

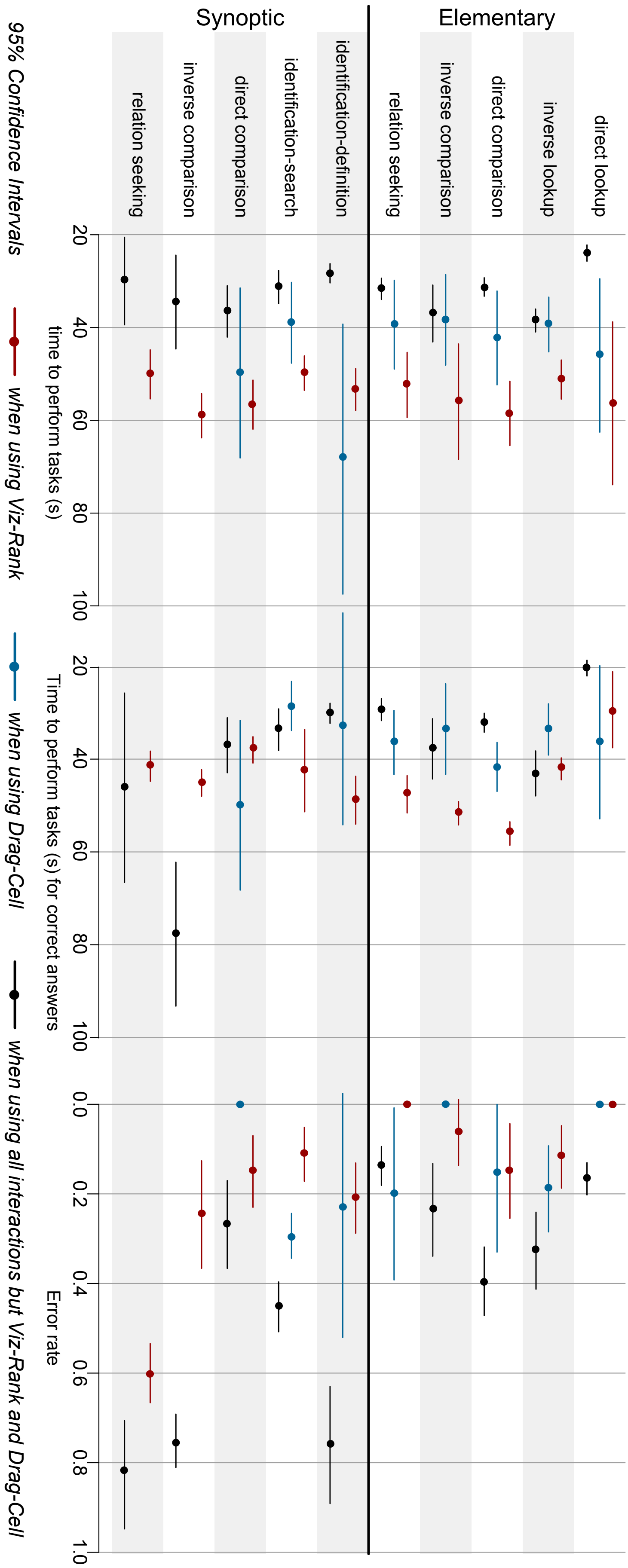



Figure 4.7: Time and Error for each task category, by performed interactions.

Table 4.1: Significant results for answers based on DRAG-CELL (*DC*), VIZ-RANK (*VR*), and exclusively other interactions (*O*). We analyzed the *Error* for all tasks and the *Time* for correct answers only.

| Factor | Task category | F | p | Pairwise mean comparisons | Mean DC | Mean VR | Mean O |
|---|---|---|---|---|---|---|---|
| *Error* | elementary inverse lookup | $F_{3,401} = 3$ | * | $DC < O$ & $VR \ll O$ | 0.19 | **0.11** | 0.34 |
| | synoptic identification-definition | $F_{3,406} = 3$ | * | $DC < O$ & $VR \lll O$ | 0.22 | **0.21** | 0.75 |
| | synoptic identification-search | $F_{3,423} = 5$ | ** | $VR \lll O$ | 0.28 | **0.11** | 0.45 |
| | synoptic inverse comparison | $F_{2,143} = 27$ | *** | $VR \lll O$ | / | **0.25** | 0.77 |
| | synoptic relation seeking | $F_{2,186} = 6$ | ** | $VR \ll O$ | / | **0.61** | 0.82 |
| *Time* (correct answers) | elementary direct lookup | $F_{2,337} = 9$ | *** | $O \ll DC$ & $O \ll VR$ | 38s | 33s | **19s** |
| | elementary inverse lookup | $F_{3,241} = 5$ | *** | $DC \ll VR$ | **36s** | 43s | 45s |
| | elementary direct comparison | $F_{3,206} = 21$ | *** | $O \lll VR$ | 42s | 58s | **33s** |
| | elementary inverse comparison | $F_{3,74} = 6$ | *** | $DC \ll VR$ & $O < VR$ | **36s** | 51s | 39s |
| | elementary relation seeking | $F_{3,261} = 12$ | *** | $DC \ll VR$ & $O \lll VR$ | 35s | 48s | **30s** |
| | synoptic identification-definition | $F_{3,226} = 22$ | *** | $O \lll VR$ | 32s | 49s | **31s** |

* for $p \leqslant 0.05$, ** for $p \leqslant 0.01$, *** for $p \leqslant 0.001$; Welsh two sample t-test: $x < y$, $x \ll y$, $x \lll y$.

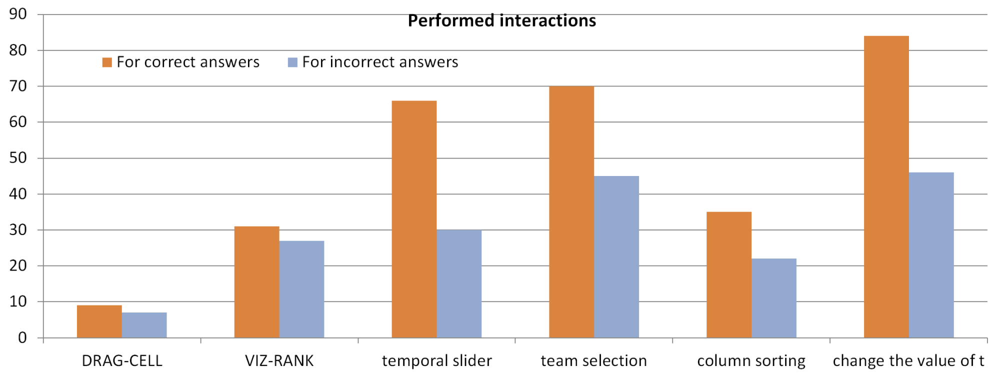



Figure 4.8: Percent of answers for which interactions were performed at least one time, for correct and incorrect answers.

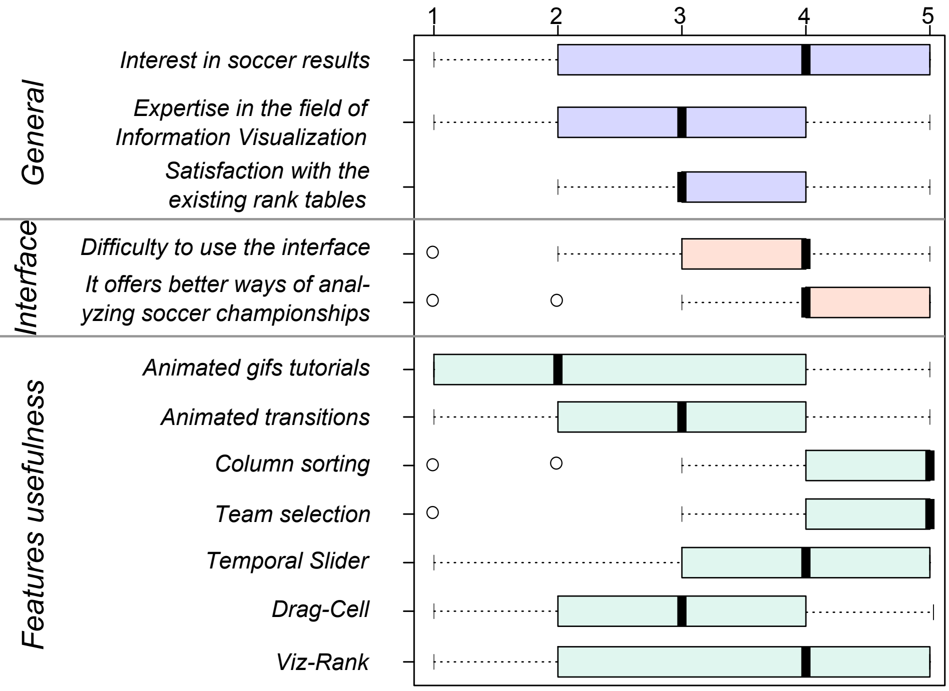



Figure 4.9: Answers to the questionnaire, on Likert scales $[1-5]$.

## 4.5 DISCUSSION

### 4.5.1 *Observations from Experiment and Main Findings*

Soccer enthusiasts with heterogeneous backgrounds in Information Visualization discovered and performed *À Table!* interactions. Our investigation confirms **H1**, although users rated the interface as difficult to use. We also had to provide an incentive in order to engage participants and make them complete all the tasks.

Some participants commented that it was difficult to discover the interactions. For example, one participant wrote: *"I think it has a low discoverability for the graph and the drag in the cell"*. DRAG-CELL and VIZ-RANK are relatively advanced techniques, performed directly on the table, to let users focus on their task. For such techniques, discoverability is a well known problem (as noted by Bederson [Bederson, 2004]). We provided a side tutorial for beginners to learn by example. However, the results showed that almost all participants discovered both techniques by exploring the table, without using this side tutorial and they often accidentally triggered the interactions the first time. In fact, both Drag-Cell and Viz-Rank appear to be quite discoverable without specific affordance.

Answers to the questionnaire confirm the interest from soccer enthusiasts for *À Table!*: 74% of them discovered new information and 31/34 participants gave a score higher than 3 on the 1-5 Likert scale to rate how *À Table!* offers better ways of analyzing soccer championships. For example, one participant wrote: *"wish I had it for my own championship (UK)"*.

*O* is faster than *VR* for both elementary direct lookup and comparison. This is explained because choosing the appropriate value for t is enough to perform the task well and using Viz-Rank makes the user waste time. *O* is also faster than *DC* for elementary direct lookup, so we partially confirm **H2**.

For both elementary inverse lookup and comparison, *DC* has the lowest mean time and is significantly faster than *VR* and we partially confirm **H3**. We also note that for inverse lookup tasks, we did not find any significant difference between *DC* and *O* because the standard deviation for *O* was too high, although it was the slowest technique overall. Drag-Cell was rarely used, but when it was, participants performed tasks faster and with fewer errors. These results confirm that Drag-Cell is efficient to perform the tasks it was designed for.

For four synoptic tasks (identification definition, identification search, inverse comparison, and relation seeking) *VR* has significantly less error than *O*, participants making respectively 3.6, 4.1, 3.1 and 1.3 times less error using Viz-Rank. We did not find any significant difference between *DC* and *VR* for these tasks but the mean error is the smallest for *VR* and we partially confirm **H4**. We also note that *DC* has less error than *O* for the identification definition task.

*VR* is the slowest for five out of the six tasks for which we were able to analyze time, with *VR* being almost twice as long as *O* for several tasks. For half the tasks with significant difference, *VR* is slower than both *DC* and *O* and we partially confirm **H5**. We explain this result because participants may not be familiar with line charts. However, it is not clear if this is a general issue, if this is in the context of rank visualization or because of the short apparition of the chart.

Participants used Viz-Rank for complex synoptic and relation-seeking tasks that are extremely painful to perform without a dedicated technique. The technique involves fewer errors but requires more time. However, because journalists cannot afford to publish incorrect data, the technique must be accurate, even at the cost of a slight increase in completion time; comparing time between techniques makes sense when the techniques have similar error rates, which is not the case.

The main takeaways from our experiment are as follows:

- *Basic interactions remain essential, fast to operate and easy to learn for basic tasks.* However, some tasks are almost impossible to perform using only these features (up to 82% of error) and advanced interaction techniques are required.
- *Interactive techniques are crucial to explore temporal ranking tables*. The more interactions are used, the more the task is performed correctly (Figure 4.9).
- DRAG-CELL *is fast and error-less for the tasks it was designed for (inverse tasks)* but almost never used for other task categories.
- VIZ-RANK *is accurate and well-suited to synoptic tasks and all tasks can be performed using it.* However, it is slow to operate and requires user's basic knowledge in Information Visualization.

Finally, both DRAG-CELL (for elementary inverse lookup and comparison) and VIZ-RANK (for four out of the five synoptic tasks categories) makes it easier to perform the tasks they were designed for. Moreover, several of these tasks were almost absent from the task analysis from soccer articles but were estimated as highly relevant by the expert. We may expect that by providing such interaction techniques as DRAG-CELL and VIZ-RANK, the analysts will be able to perform tasks they can not perform today and therefore enhance the quality of the insights and statistics they can retrieve from the data.

### 4.5.2 *Applicability Beyond Soccer and Limits*

*À Table!* is seamlessly applicable to other sport ranking tables showing team statistics. We now describe *À Table!* applicability to two types of ranking tables from different domains and discuss possible issues and limits.

ACADEMIC RANKINGS Since 2003, the shanghai:2013 University [Consultancy, 2013] publishes every year an updated Academic Ranking of World Universities. It ranks the top $|S| = 500$ institutions, with $|D| = 6$ dimensions. The main rank is computed on the number of Nobel Price or Fields Medal winners from the institution, among other criteria. Top institutions like Harvard and MIT are respectively ranked first and second, and tracking them over time can be done visually. *À Table!* would be efficient to browse the top-k, $k = 20$ ranks for which permutations are important but mostly remain on the same screen. However, some issues occur for ranks lower than 100 which are not all visible without scrolling and contain ties. Furthermore, it only ranks the top-500 universities every year, meaning that new institutions may appear or disappear over time, making their tracking difficult.

SORTING ALGORITHMS EXECUTION Sorting algorithms, such as quicksort or bubble sort, perform many permutations until they end in a final result. The number of permutations varies according to the size of the dataset $|S|$. The type of permutations varies according to the algorithm itself. Existing works already explored the visualization of execution steps[4] which enable a non-expert to grasp the underlying sorting strategy. *À Table!* would be useful to inspect executions and to reach a specific execution step or time. Such an investigation of algorithms—or any ordered dynamic process—can be used in a variety of contexts, ranging from educational purpose to algorithms optimization.

LIMITS Some real-life rankings may not be immediately compatible with current *À Table!* design. It is already challenging to visualize and interact with rankings with partial ranks, missing data during time intervals, new dimensions, etc. In contrast, tables do justice to such rankings by creating empty rows or additional columns to represent missing data. It may not be optimal but it has the merit of being a consistent representation.

### 4.5.3 *Design Implications*

Adding novel interactive techniques to legacy techniques, such as ranking tables, implies many design considerations. We only scratched the surface of making them fully effective, and we share some lessons we learned on their discoverability and learnability. Discoverability is mainly connected to the affordance of the first interaction to **start** the technique. We did not use any for simplicity, and considered natural to change values by dragging them up and down. Some codes for affordance are already available, such as Bret Victor's [Victor, 2013b] use of dashed lines to show draggable values. Learnability can be supported in many ways. The tutorial we provided during our experiment was probably not sufficiently contextualized, explaining why most of the participants discarded it. It seems that it was natural for users to click and drag cells as they are their primary interest. They may have acquired this knowledge elsewhere and applied it to *À Table!* by associativity [Anderson, 2000]. Line charts provide visual guides, but require space and time to appear. Exploring hybrid features, such as dragging values with a visual overlay showing upcoming and past values, might be the best of the two worlds but needs more investigation. Still, regular dragging can be left as an expert-feature [Bederson, 2004].

4 http://www.youtube.com/watch?v=m1PS8IR6Td0

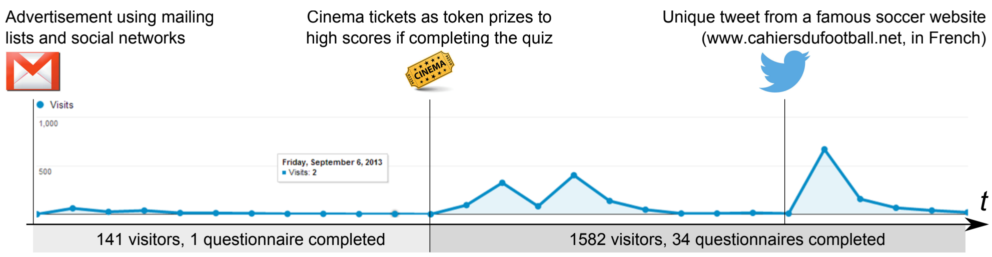



Figure 4.10: User engagement for the online experiment of *À Table!* over a three weeks period. The number of visits changes after each of the three events: advertising the url, offering token prizes, and a tweet.

## 4.6 CONCLUSION

This chapter presented *À Table!*, an enhanced ranking table with improved temporal navigation, by combining two novel interaction techniques. Initially, our goal was to better support time exploration in ranking tables through interaction, as the temporal dimension is crucial to understand soccer championships. Our evaluation shows that the techniques efficiently support important temporal tasks, currently not supported by standard ranking tables. This paves the way for efficiently introducing advanced visual exploration techniques to millions of soccer enthusiasts who use tables everyday, as well as other application domains which use ranking tables.

We also thought of designing novel navigation techniques to include in *À Table!*. One of them is using a table cube metaphor [Bach et al., 2014], with a transient 3D animation similar as Rolling the Dice [Elmqvist et al., 2008a], to change a time t or a dimension d step-by-step. This type of transition provides a consistent temporal navigation paradigm—adequate with tables which are also squared—and perfectly fits the table layout.

By *integrating* a line chart into a table, we propose an alternative presentation of the data when the user needs it in order to better support different tasks [Casner, 1991]. We now discuss the issues raised about engagement, discoverability of the techniques, both efficient and fluid interactions, and a tradeoff between congruence and versatility.

### 4.6.1 *User Engagement*

Over the three-weeks period of the experiment, we experienced engagement issues from the visitors. Figure 4.10 shows the number of visitors over this period. We quickly realized that only advertising the webpage url would not be enough to engage people. Thus, following Mechanical Turk's incentives strategy, we offered cinema tickets as token prizes. Immediately, almost one thousand people visited the website. A few days later, a famous french soccer website wrote a tweet about the website, and it resulted in a new bump of visits.

However, analyzing the data clearly showed that the number of visits is not correlated to the quiz completion. Although a large number of visitors accessed the webpage right after the tweet from the soccer website, the largest number of completed quiz and questionnaire occurred after the token prizes advertising. The conclusion from this experience is that incentives are almost mandatory to engage users. As expected, soccer enthusiasts were eager to explore the data and spent time on the page. However, without any incentive, they were not willing to complete the quiz and even less interested in providing questionnaire answers and feedback.

▷ Incentives strongly engage users, and other incentives than a payment can be used.

### 4.6.2 *Interaction Techniques Discoverability*

Because we did not run a controlled lab study, we had no way of ensuring that the participants would understand and learn the interaction techniques. To facilitate learning, we added a pointer to a help page providing instructions and examples illustrating the interactions. We also embedded animated images in the bottom right of the interface as shown Figure 4.3, assuming that the animations would draw the attention of the participants to these images and help them learning by example.

However, analyzing the logs showed that less than 1% of the visitors visited the instructions page, and results from the questionnaire answers showed that participants did not find the animated images useful. As a result, some of the participants commented that it was difficult to discover the interactions. Drag-Cell and Viz-Rank are relatively advanced techniques, performed directly on the table, to let users focus on their task. For such techniques, discoverability is a recognized problem [Bederson, 2004].

Nonetheless, participants used both techniques to perform the tasks. According to questionnaire answers, almost all participants discovered both techniques by exploring the table without using the provided tutorials and often accidentally triggered the interactions the first time. Our guess is that interaction techniques can be discoverable by themselves, without the need of hints and tutorials. Indeed, for this experiment, it seems that users did not want to spend time reading instructions or looking at animated images but preferred to interact randomly with the table until they trigger unexpected interactions.

Finally, we believe that the discoverability of interaction techniques is directly linked to the level of engagement of the users: if engaged, they are much more eager to explore the visualization and discover the available interaction techniques. However, discoverability remains problematic and an important challenge of interaction and interface designs; more work is needed in this direction to assess what makes an interaction technique discoverable, and what is the value of discoverability.

▷ Interaction techniques discoverability is an important and open challenge that may be related to user engagement.

### 4.6.3 *Efficient fluid Interaction*

Here we refer to the summary of principles, benefits and challenges summarized in Table 2.1. We designed novel interactions following Bederson [Bederson, 2004] principles to preserve the user's flow, to focus on tasks and prevent interruption. Providing an invisible interface (P16) and using smooth transitions between states (P15) give the user a sense of control. The non-interfering interface (P10) with no error message (B4) and rewarding interaction using animated transitions (P17) make the system playful and enjoyable. This reduces user's anxiety (B6), as indicates the feedback from the participants to the crowdsourced evaluation. Despite the fact that seamless and playful interaction may conflict with efficiency, we showed that performance increases using these two interaction techniques (C2) by manipulating directly the objects of interest instead of additional widgets that are not efficient (C15). In the case of temporal tables, difficulty lies in the task domain (C3) and allowing to perform all the temporal tasks is difficult.

### 4.6.4 *Congruence vs Versatility*

Standard widgets are not congruent to most of the tasks. A slider or a dropdown list are congruent to elementary direct lookup tasks only. For example, analyzing or comparing trends, finding the time when something happens, and comparing two values at different times are tedious to perform.

Viz-Rank and Drag-Cell are two-dimensional interactions for three-dimensional information spaces (C16). Both techniques have a low degree of spatial indirectness (P11), but a different degree of congruence.

Drag-Cell is a congruent interaction and provides a natural mapping to the task (P8), despite having a high degree of temporal indirectness (P12) that we compensate using a preview of the result while performing the interaction. Indeed, to perform *inverse* tasks, interacting with the *values* domain has a much lower articulatory directness (P7) than interacting with the *time* domain.

Conversely, Viz-Rank is not highly congruent to the tasks it is designed to perform for several reasons. First, it also has a high degree of temporal indirectness (P12) because of the stagged transitions between states. Second, it aims at making easier a wide range of tasks. Each task category having a different articulation, it is difficult to imagine a unique interaction that would be congruent to all task categories.

On one hand, Drag-Cell is highly congruent but dedicated to a small subset of tasks only. On the other hand, Viz-Rank is less congruent but dedicated to a much wider range of tasks. This raises the tradeoff between congruence and versatility of the interaction technique.

# 5 TIME NAVIGATION IN DATA GRAPHICS BY DIRECT MANIPULATION

Ranking tables—tables ordering objects according to one criteria—are not limited to soccer: sports in general is an obvious generalization, but for example world universities are ranked according to their academic performance every year, and websites are periodically ranked according to their popularity. Updating ranking tables over time involves the vertical permutation of rows. When navigating over time, animated transitions can facilitate the understanding of the update, but tracking multiple moving elements—even with only one degree of freedom—is difficult and time consuming.

Surprisingly, temporal navigation in such ranking tables remains difficult. In Chapter 4, we showed that standard widgets such as sliders are not optimal for the temporal exploration of rankings, and we advocated interacting with the value domain (i. e. the ranking or any value in the table) rather than with the time domain for several important temporal tasks. Similarly to navigation, understanding temporal changes remains difficult as animated transitions produce visual clutter and overlaps. Using adjacent tables [Gratzl et al., 2013] or small multiples for each time unit prevents the overlap, but at the expense of using more screen real estate and scaling down the table, thus reducing the legibility of symbols, and and at the cost of not conveying trends.

We first propose to improve animated transitions in ranking tables by 1) displaying the trajectory of row permutations as a ranking chart; and by 2) letting the user drag objects following the chart, to control transitions pace and direction. This technique preserves the advantages of data tables to inspect and compare values, while adding the benefits of ranking charts to view trends over a longer period of time. We explore several design variations for four types of rankings: soccer teams, countries, universities and sorting algorithms rankings.

In the second part of this chapter, we discuss the applicability of this technique beyond ranking tables, as a general principle for direct and informative manipulation of temporal data graphics. We explore the design space of the moving objects' trajectory, which is a 2D path with motion over time, and the interaction model, which is the series of states and transitions between the graphics' temporal snapshots.

This chapter addresses the following questions:

- ▷ What makes a time-navigation interaction technique direct?
- ▷ How to seamlessly interact with time-dependant data graphics to navigate in time?
- ▷ How to represent the trajectories of objects of interest?
- ▷ How to control animated transitions using trajectories?

## 5.1 RELATED WORK

This chapter is related to direct manipulation principles, time navigation, and interactive navigation in graphical interfaces.

### 5.1.1 *Temporal Tasks*

We refer to the task taxonomy for temporal data by Andrienko and Andrienko that we detailed in Chapter 4 [Andrienko and Andrienko, 2005]. In this previous chapter, we showed that traditional time sliders, designed to quickly reach a given time, are efficient for direct tasks. However, they are not efficient for synoptic and comparison tasks, requiring alternative representations and interaction techniques. Actually, for most of the tasks, a time slider is not as direct as it should be, being tedious to use for performing important tasks. In the following subsection, we analyze what makes a good interaction technique for time navigation and we emphasize the weaknesses of the time slider.

### 5.1.2 *Interaction Directness*

In this subsection, we refer to the direct manipulation principles listed in Chapter 2. A time slider has a high semantic directness in both the gulf of execution (P5) and the gulf of evaluation (P6), a high articulatory directness (P7), and it provides a natural mapping to the task (P8, P19) only if the task consists of reaching a point in time (direct tasks). It usually provides immediate feedback (P9) and is not obtrusive (P10).

However, a time slider has a high spatial directness (P11) because it requires to split attention between the slider (the instrument) and the visualization (where the objects of interest are), but it usually has a high temporal directness (P12), when the content of the visualization is immediately updated. It has a low degree of integration (P13) because only one spatial dimension of the 2D mouse is used, and a low degree of compatibility P14 because translating the slider thumb along one spatial dimension makes the objects of interest in the visualization move in the 2D space. The slider is also a separate interface component, reducing the interface invisibility (P16), and by default it does not respect P15 and P17. When the objects of interest are directly manipulable, navigating using the slider requires an interaction modality change, conflicting with P18.

In Chapter 4, we proposed DRAG-CELL and VIZ-RANK, two alternative techniques for temporal navigation in ranking tables. Figure 5.1 summarizes each technique according to the direct manipulation principles. The time slider does not respect most of the principles. DRAG-CELL and VIZ-RANK respect more principles, but suffer from a low temporal directness (P9, P12) and a low degree of compatibility (P14). To investigate how DRAG-CELL and VIZ-RANK can be merged into a 'more direct' navigation technique, we now explore related work on navigation using trajectories, animated transitions, and spatial navigation in user interfaces.

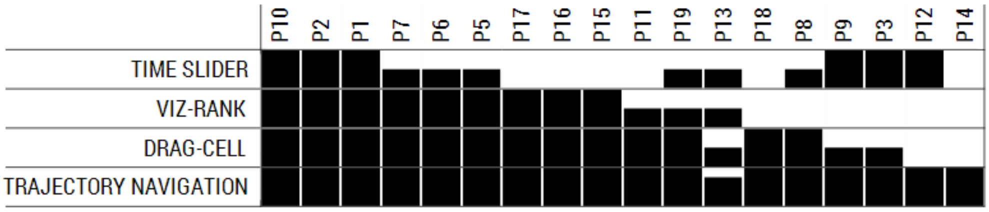



Figure 5.1: Time slider, Drag-Cell and Viz-Rank according to direct manipulation principles. The last row of the table shows the high directness of the temporal navigation by direct dragging of the objects of interest.

### 5.1.3 *Trajectories Navigation*

A trajectory is the path of a moving object over time. It is visually encoded as a line, each point corresponding to an actual information on the object's position. Trajectories have been used for browsing visual objects in videos to access a specific time frame [Dragicevic et al., 2008]. The idea is to directly drag visual features in videos as objects of interest, instead of using a time slider. However, the trajectory does not always represent every single time unit, for example if an object stays several time steps at the same position. Those shortcomings can be tackled either with spatial distortion to spatially encode time with an extra path, either using color to indicate the pause on the trajectory [Karrer et al., 2012]. More generally, showing the trajectory of elements is relevant to contextualize the animation of a set of objects [Rosling, 2009], and to better support pattern identification [Robertson et al., 2008]. However, it is rare to have visible trajectories by default. They are usually used as visual cues to show the latest states of elements, which may fade away after a couple of seconds [Huron et al., 2013]. Trajectories can also be triggered by an interaction to preview a path of moving object or viewport such as spatial direction [Dragicevic et al., 2008] and single [Baudisch et al., 2003; Moscovich et al., 2009] or multidimensional route [Elmqvist et al., 2008a].

### 5.1.4 *Animated Transitions Control*

Animated transitions in user interfaces facilitate understanding changes from one state to another [Hudson and Stasko, 1993]. They are usually empirically designed, but can eventually be sophisticated to connect two different rendering types [Brosz et al., 2013; Dragicevic et al., 2011b; Heer and Robertson, 2007]. Using traces or trails of moving elements can help the user to follow animated objects [Brandes and Corman, 2002; Huron et al., 2013; Robertson et al., 2008; Wolter et al., 2009].

Transitions are usually triggered by an action refreshing the interface, and users rarely have control over them. However, for tasks where comparison is important, the user may take over on its timing. The time slider is the default navigation widget [Chevalier et al., 2010], but an an overview of the dataset can be used instead [Elmqvist et al., 2008a], at the cost of introducing spatial indirectness.

### 5.1.5 *Spatial Navigation in User Interfaces*

Visualization tools usually follow Shneiderman's mantra, *Overview first, Zoom and Filter, then Detail on Demand* [Shneiderman, 1996]. Thus, they require navigating into the viewport [Yi et al., 2007] through scrolling, panning, and zooming that do not provide guidance to explore large or complex spaces. Many iterations on those navigation techniques aimed at tackling this issue. For example, content-aware scrolling [Ishak and Feiner, 2006] creates a virtual navigation path to scroll a two-columns document the way a reader reads it, while still using the scrollbar. When the user's path is unknown and many choices are available (e. g., when dragging icons on a large screen [Baudisch et al., 2003]), showing the potential paths is useful and sometimes necessary. Similarly, data-oriented visuals act as guidance, such as showing the existing connections of a graph as a path to improve navigation tasks [Moscovich et al., 2009]. Because dragging over a path requires substantial steering efforts, a loose coupling is often tolerated to avoid having to drag along the exact path [Dragicevic et al., 2008].

When the viewport is fixed, some invisible objects may be unreachable. An alternative to spatial navigation consists of grouping the remote elements close to the immobile object of interest [Baudisch et al., 2003; Moscovich et al., 2009]. Other alternatives [Irani et al., 2006] consist of displaying invisible objects on the frame of the viewport [Kwon et al., 2011] and showing dynamic insets at the periphery of the viewport [Ghani et al., 2011]. Even subtle visual cues can give a sense of navigation by indicating what will happen next in an informative way, both for static widgets [Willett et al., 2007] and using dynamic visual cues when dragging values (Drag-Cell, Chapter 4).

### 5.1.6 *DimpVis*

Direct Manipulation has already driven the design of time navigation techniques such as video browsing in DimP [Dragicevic et al., 2008] and scientific visualization [Wolter et al., 2009].

More recently, and in parallel to this work, DimpVis [Kondo and Collins, 2014] applies direct manipulation to time navigation in abstract data graphics. They apply the browsing strategy used in DimP to interact with time-dependent data graphics by manipulating directly the objects of interest of the visualization instead of a time slider. This chapter is closely related to DimpVis and we believe that both works are complementary as they emphasize different aspects of time navigation by direct manipulation.

They found that their technique was subjectively preferred by the participants than the time slider and small multiples, but the results from the experiment did not lead to significant conclusions in terms of efficiency. In this chapter, we focus on exploring the design space of transient trajectories and how they can convey more information than only the motion path of the objects. We also explore alternative dragging strategies, depending on the visual representation and trajectories properties, and we propose a generic interaction model that can be implemented with little effort to augment visual representations of time-dependent data graphics.

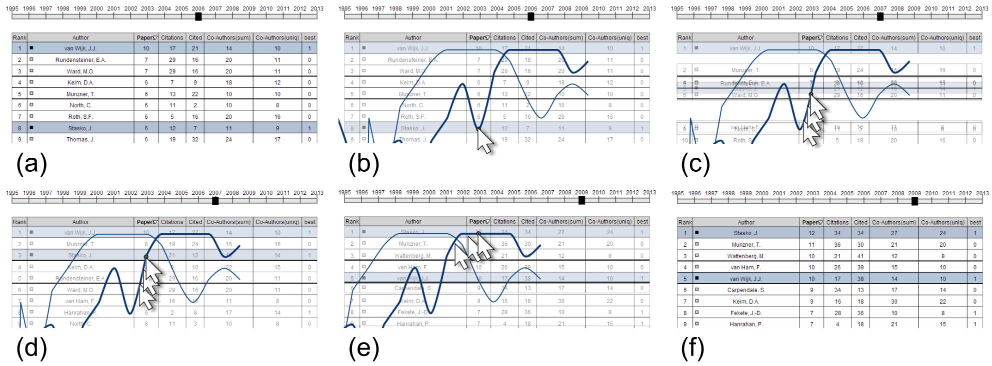


Figure 5.2: Temporal navigation using a transient line chart. The ranking table and the transient line chart have a similar vertical layout.

## 5.2 improving temporal navigation in ranking tables

We propose to merge Drag-Cell and Viz-Rank into a transient line chart to keep the best of both techniques and navigate simultaneously both in the values domain and in the time domain. In particular, we carefully consider the principles that these techniques do not fully respect:

P3 The interaction must be reversible and the effects on the objects of interest must be immediate.
P8 The gesture should be a drag, not a click.
P9,P12 There should be little to no delay between the action and the visualization updates.
P13 The two spatial dimensions of the mouse should be used.
P14 The modified object of interest must follow the mouse cursor.
P18 There must be no mode change, both the table and the motion trajectory cohabiting simultaneously.
P19 The direct manipulation of an object of interest must reflect the performed task.

### 5.2.1 *Motion-Trajectory Navigation*

The interaction technique we propose, temporal navigation using a transient motion trajectory, is illustrated Figure 5.2 for the Infovis citations dataset [1]. (a) The user first selects the rows to analyze, here John Stasko and Jarke Van Wijk. (b) Pressing the mouse on the cell representing the number of papers of John Stasko makes the line charts associated to the selected rows appear, with a thicker line for John Stasko; a circular cursor appears on the thicker line chart. (c–e) Dragging the mouse along the trajectory makes the line chart move right or left to explore all the possible values over time. (f) Releasing the mouse drag makes the transient line charts fade out and the table goes back to a stable layout.

An important property of the technique is the similar layout for the table and the line charts: the vertical position of a row is always the same as the

1 http://www.cc.gatech.edu/gvu/ii/citevis/

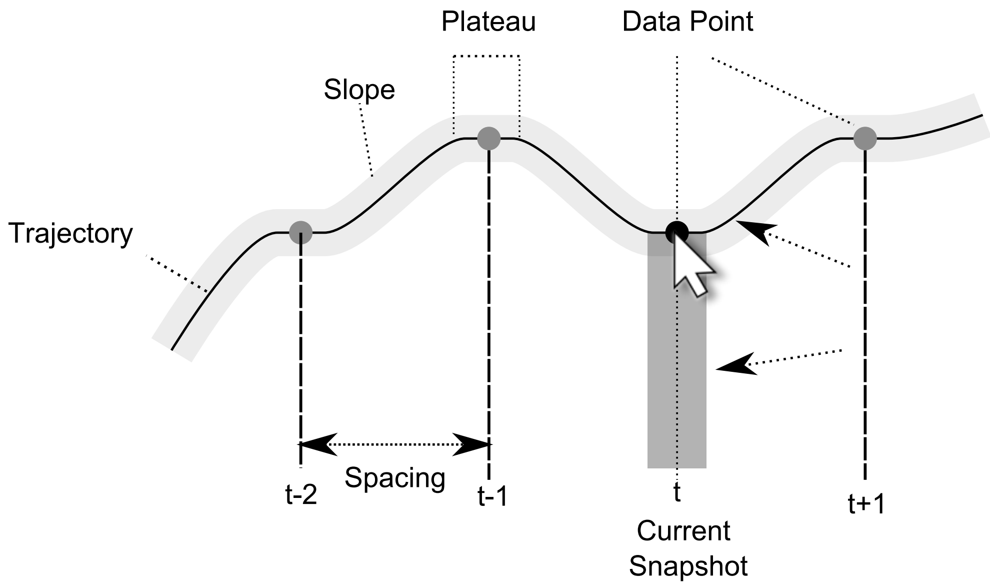



Figure 5.3: A trajectory is the visual connection of data points from graphical snapshots, spatially ordered by time (from left to right).

vertical coordinate of the cursor on the corresponding line chart. By controlling the pace of the animation, the user can carefully explore the data—e. g., look for intersections—while observing row permutations. For example, in Figure 5.2, the cursor on the linechart appears at the same position as the clicked cell was (b). When dragging along the trajectory, the associated row vertically follows the cursor (c-d-e), and when releasing the drag, the row of interest is at the same vertical position as the cursor was (f).

### 5.2.2 *Trajectories Design Space*

In this section, we detail the design variations for motion trajectories to establish their design space. The trajectory is the graphical representation of an object of interest's path over time, and the trajectories design parameters are illustrated Figure 5.3. The path is constructed by the visual connection of data points, which are the positions of the object at each snapshot of the updated graphic. An efficient trajectory design facilitates navigation for the user, to let her focus on the values of the dataset and her task rather than the current time. For clarity, we consider the trajectory of elements moving in one direction: from left to right; with the vertical axis encoding the data values. We also only consider simple continuous trajectories at this point (i. e. no loop or discontinuity).

The *spacing* is the distance separating two snapshots. The spacing ensures that the animation timing from one update to another is well adjusted (not too short or too fast). The design for spacing is similar to the choice of a x-scale for a line chart, and similarly very long time series may be problematic.

The *path* of the trajectory is constructed by *curve fitting*, which consists of finding a mathematical function that has the best fit to a series of data points.

Curve fitting involves either exactly fitting the data points (interpolation) either approximately fitting them (smoothing). Both strategies result in a chart similar as a line chart (Figure 5.3) with possible loops. The main drawback of smoothing the path is that the trajectory may not be faithful to the data and would make more difficult to perform tasks involving values retrieval. However, using such lines can be useful for relative dragging, as we will discuss later.

A *plateau* is an optional area surrounding the current data point where the line becomes horizontal. In terms of animation, a plateau's length is equal to the duration of the animation around the point. Other examples of plateaux are Slope Graphs [Tufte, 1986], where plateaux themselves are symbols, values or rankings, and where a single line connects plateaux to each others. The maximum width for the plateau is the time unit, and the minimum is 0. Plateaux can either result from the interpolation functions either be affected by changing the spacing between data points and adding intermediary control points.

The *slope* is the steepness of the curve, which is the complement of the plateau's width. Steepness produces a fast transition. Inversely, a flat slope will provide a slower animation. Slopes are always visible, even if infinite. They can be designed to ease the connection with the plateaux (and plateaux themselves participate to the easing effort) but then the separation with the interpolation becomes blur [Dragicevic et al., 2011a].

*Decorations* can provide visual aid to indicate attributes related to data points position. One fundamental design decision in our context is to visually represent (e. g., using circles) actual data points on the trajectory to make them more salient than interpolated ones. Also, as many trajectories may cohabit, using semi-transparent color to only show context [Dragicevic et al., 2008] or create a trail effect of latest values [Brandes and Corman, 2002; Huron et al., 2013] allows filtering objects of interest. Another valuable design decision is to show the direction towards which to drag [Baudisch et al., 2003]. Further decorations such as axis and labels can be used (e. g., displaying each time step [Kondo and Collins, 2014]), but may clutter the graphics' snapshot.

The path of the trajectory plays a role not only in the visual appearance of the representation, but also in the animation behavior. Thus, the design decisions depend on the final rendering as well as the dataset and tasks, as these may impact the interaction fluidity, the engagement from the user, and the efficiency of the interaction. For example, a *linear* interpolation corresponds to a continuous animation with no delay: only the speed of the animation changes according to the angle of the current segment and is constant as long as another segment is not reached (i. e. animation speed is piecewise constant). Or, with a *step* interpolation (maximum width plateau), the slope of the line is either null (i. e. horizontal) or infinite (i. e. vertical). In terms of animation, it corresponds to no speed at all: the horizontal segments of the line encode the duration of each snapshot, and the vertical ones the duration of the switch to another point, which is 0.

| Case Study | #Rows | #Dims | #Time | Plateau | Prov. | Land. | Interp. | Halo |
|---|---|---|---|---|---|---|---|---|
| Soccer Teams | 20 | 10 | 38 | Small | Circle | Top-3 | Linear | No |
| Countries | 134 | 3 | 58 | None | None | None | Poly | No |
| Universities | 50 | 5 | 7 | Large | None | Top-10 | Poly | No |
| Sorting (small) | 20 | 8 | 90 | Large | None | None | Linear | No |
| Sorting (large) | 400 | 8 | 4252 | Large | None | None | Linear | No |
| Citations | 50 | 8 | 18 | None | Circle | None | None | Yes |

Table 5.1: Summary of the case studies design variations and techniques characteristics: number of rows (entries), number of columns (dimensions), number of time steps, plateau size, causality, landmarks, type of interpolation, and presence of a halo.

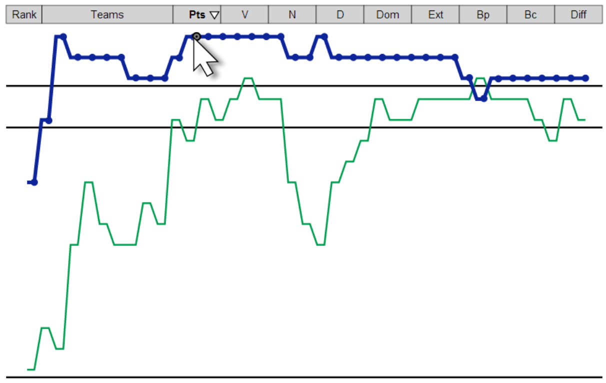



Figure 5.4: Two soccer teams fighting for the top-3 positions.

### 5.2.3 *Case Studies*

We built a web prototype using SVG and D3 [Bostock et al., 2011] to explore the design space and test design variations for ranking charts and dragging sensitivity parameters. Because it is not clear yet which design setting is efficient or not, we made all of them available as parameters embedded in the URL. We ended up with 15 parameters, with in average 3 to 4 variations for each. Table 5.1 shows examples of configurations. The prototype is available online [2] and will be released as an open source project.

We selected five temporal rankings as case studies: *soccer teams* ranking as standard ranking with landmarks; *countries* ranking for many rows and time units; *universities* ranking for missing columns; *algorithm sorting* to demonstrate the scalability of the technique in rows and time intervals; and *IEEE Infovis citation* data to explore data causality. Table 5.1 shows a summary of the rankings' characteristics and the techniques' design variations.

2 http://90plan.ovh.net/~lafrancep/transient/

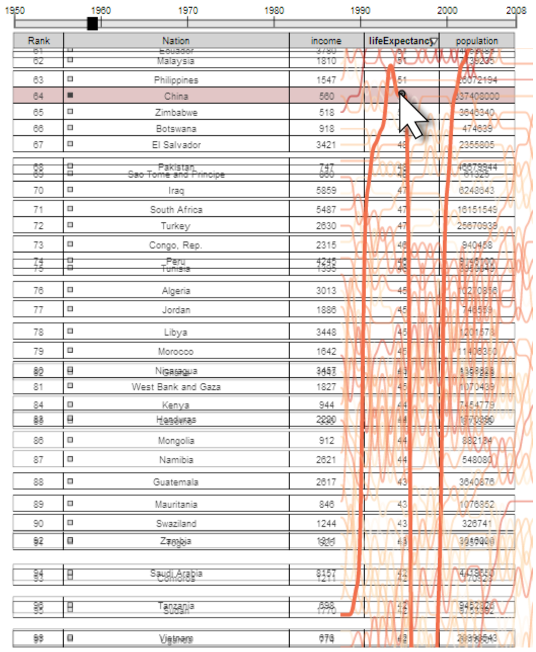


Figure 5.5: Countries ranking by life expectancy.

#### 5.2.3.1 *Soccer Ranking Tables*

Soccer ranking tables present team performances during a championship, and are updated after each game. Rows are teams and columns are the number of points, to compute the rank. Other columns contain related statistics, such as goals scored or conceded, but are not used to decide which team wins the championship. Soccer championships often involve 20 teams, resulting in 38 games where the average time unit is a week.

We used a linear interpolation and small plateau, a relatively standard representation for ranking charts. We extended the technique in two directions. First we added small plain circles as data points representations, for every value (i. e. every game). This is helpful when a team has a flat curve, to know how many games in a row were played during this time period. A second improvement was to keep the landmarks of the original table, which are the three horizontal lines in the ranking charts. It shows which top-3 teams are qualified for the European cups, and which bottom-3 ones are demoted to a minor league. We also used semantically resonant colors matching: the color of the team logos. Figure 5.4 immediately shows teams converging to the higher ranks, with a brief ranking permutation.

#### 5.2.3.2 *Countries Ranking*

In economics, it is frequent to rank countries according to various indicators to compare the efficiency of national policies or international aid use. Figure 5.5 shows 134 countries over the last 58 years [Rosling, 2009], ranked by life expectancy. The highlighted chart is the one for China, and the deep valley is due to the Great Chinese Famine (1958-1962). Such as variation is

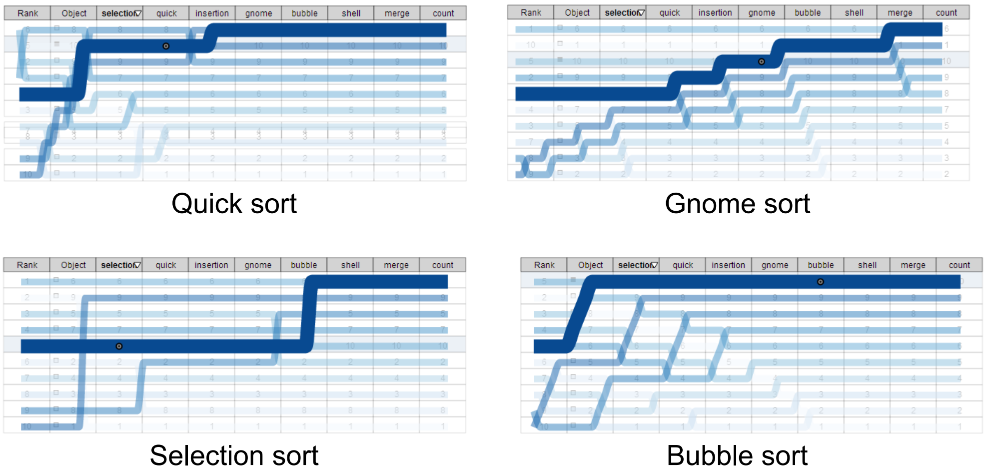



Figure 5.6: Examples of sorting algorithms: Quick, Gnome, Selection and Bubble sort.

a starting point for further exploration by either selecting countries with similar behavior or analyzing other dimensions.

Because the time period is long, we designed a compact ranking chart to facilitate the exploration. We kept as background both the table and the other rankings values to provide context. Because the number of countries is important, the table does not fit in the page and requires some scrolling. Two alternatives can solve this issue: either using a ranking chart that fits and stays in the page, and the table moves up and done to follow the trajectory; or the line chart is of the same height as the table and an automatic page scrolling is activated.

#### 5.2.3.3 *Universities Ranking*

Shanghai University publishes every year a ranking of universities, according to such criteria as the number or articles and the number of Nobel Price or Fields Medal winners from the institution [Consultancy, 2013].

We started by focusing on the top-50 universities, and for the period 2007–2013. We used the same design as for soccer ranking tables, with the top-10 line as a landmark. The major design limit was to account for the missing universities as some appeared and disappeared from the top-50. This was done by connecting the rankings to the lower limit of the table universities. Expending the data to the complete ranking requires novel design challenges: dealing with rankings with ties, and scrolling very long lists of rankings (similar as with countries). Finally, an open design issue is how to use 2003–2006 data where dimensions are different from 2007–2013?.

#### 5.2.3.4 *Sorting Algorithms Execution*

A sorting algorithm is a series of instruction to progressively order elements in a list. Each instruction is a pairwise comparison, which generates a time

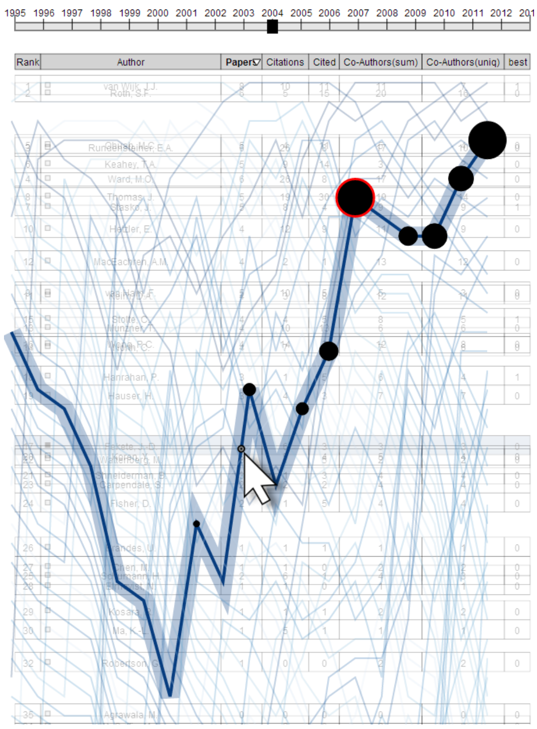



Figure 5.7: Jean-Daniel Fekete ranking over the years according to the number of citations at the IEEE Infovis conference.

unit. We implemented eight standard algorithms, and for each algorithm we computed the sorting of 10 randomly generated objects that can be sorted. We stored the rank of each object for every pairwise comparison.

We explored two sorting scenario, depending on the number of elements to sort: small (20 objects) and large (400). It respectively resulted in 90 and 4252 time units. Figure 5.6 shows the small dataset for 4 algorithms. Because such algorithms converge to a known stable state, we encoded each line color with the final rank, using a color gradient. For the small table, we used a maximum stroke width and maximum plateaux. This produces aesthetic images, which are quite similar to existing artistic renderings of sorting algorithms[3] [4]. A pedagogical application is to use this ranking navigation technique to understand or teach their execution. For the large rankings, we faced the same issues as previous case studies with numerous rows.

#### 5.2.3.5 *IEEE Infovis Citations*

For this case study, each row is an author and columns represent citation-related data such as the number of published paper and the number of citations over the years. We filtered the data to discard authors with less than five published papers in 2013. Figure 5.7 shows the number of publications for Jean-Daniel Fekete. We used no interpolation, a halo, and no plateau.

3 http://sortvis.org/
4 https://www.jasondavies.com/sorting/

Automatic page scrolling was also needed. In this scenario, the interesting information is the number of papers, thus we encoded data causality using circles whose radii encode the number of articles each year. A red outline of the circle indicates a best paper. Figure 5.7 clearly shows an increasing number of articles over the last years, resulting in a steep upward slope.

### 5.2.4 *Discussion*

From the exploration of the previous case studies, we extract design variations that should be considered when designing a visualization integrating the transient line chart interaction technique. These variations are either related to the design of the table, either general to any visualization technique with trajectory-based interaction.

TABLE DESIGN According to the dataset and/or the task, we extract the following design variations for the table:

1. Table opacity.
2. Table landmarks.
3. dynamic update of the rows position when dragging.
4. Auto-scroll for tables with large numbers of rows.

The main issue we encountered is the case of tables with large numbers of rows, not fitting in the viewport. To navigate in the table, we implemented an auto-scroll function to always have the circular cursor visible on screen. Then, the web page scrolls vertically when the cursor reaches the top or bottom third of the viewport.

One of the strength of the technique is also the layout compatibility between the table and the trajectory: despite ranking tables and ranking charts are seemingly different, at a given time, a point of the chart has the same vertical position as its corresponding row in the table of the same time. Thus, the rows of the table follow the trajectory as it is explored, and no mode switch is required (P18).

TRAJECTORY DESIGN Beyond ranking tables, we extract the following parameters that are generic to any trajectory-based interaction:

1. Trajectory interpolation type.
2. Trajectory stroke width.
3. Trajectory stroke color.
4. Trajectory plateau width.
5. Trajectory Halo.
6. Non-selected trajectories opacity.
7. Data causality representation.

Most of these design variations are straightforward to implement. The exception is data causality, which depends on the dataset and the task. For

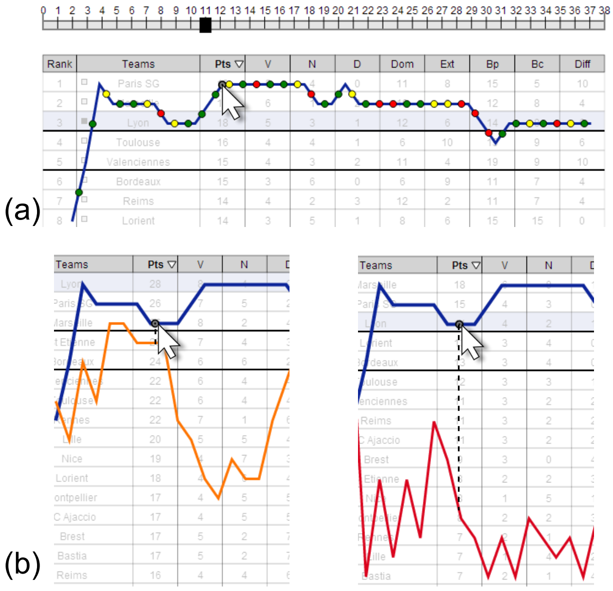



Figure 5.8: Data causality: (a) game results are encoded with colored circles; (b) the opponent of the current day is highlighted.

example, we represent data causality for citations by varying the circle radius and an outline to identify best papers; but we implemented several alternatives with other purposes. For example, when exploring the ranking table of a soccer championship, data causality can be expressed from at least two perspectives: game results and opponent identification. Each day of the championship, the rank of a team evolves according to the result at that day. We represent this information using colored circles: green for a won game, yellow for a draw, and red for a lost game (Figure 5.8(a)). This data causality encoding makes easy to locate series of won games or lost games in a row for example. Another particularity of a championship is the dependency between team results because each day, one team confronts another. One way of representing this dependency consists of highlighting the opponent of the selected team at the current time while dragging. Figure 5.8(b) illustrates this way of representing data causality. The left snapshot shows the opponent of the blue line chart at the current time highlighted (the orange line chart). By dragging vertically, the current time changes, and the right snapshot shows the opponent of the selected team at the next time step (the red line chart). This way of representing data causality is important as the length of the dashed line connecting the two line charts expresses the rank difference between the two corresponding teams at this time. In Figure 5.8(b,left), we observe that the selected team, Lyon, met Lorient (orange) while both teams were really close in the championship, respectively at the third and fourth position. We also see that the behavior of these teams diverged just after they met, with Lyon going first and Lorient loosing many positions.

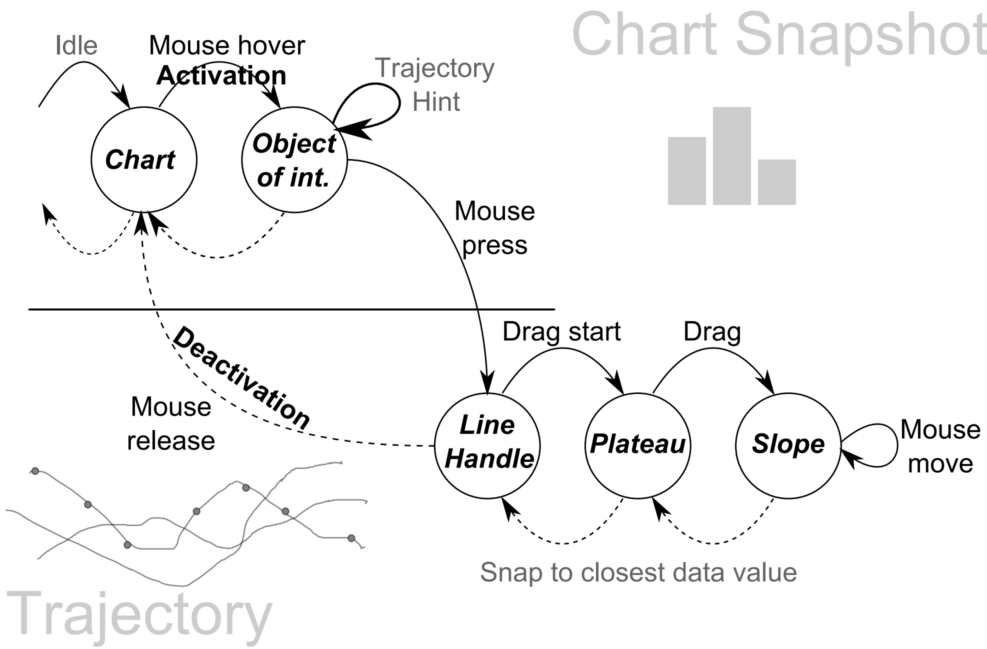



Figure 5.9: Interaction state model that shows the transition between the chart's snapshot and the line chart that serves as trajectory.

## 5.3 TRANSIENT INTERACTION FOR TIME DEPENDANT DATA GRAPHICS

While the table design variations are specific to ranking tables, the trajectory design ones can be applied to any trajectory-based interaction. In this section, we explore how the transient interaction we propose for ranking tables applies to other visual representations. We propose an interaction model for direct manipulation of time-dependant data graphics using transient trajectories, and we describe several case studies to demonstrate the versatility of the technique. We then provide in a list of guidelines and remaining challenges for temporal navigation by direct manipulation.

### 5.3.1 *Interaction State Model*

This section describes the interaction state model shown Figure 5.9. It starts with the activation phase, triggered when hovering the object of interest; and ends with the deactivation phase presenting the updated graphical visualization when releasing the drag.

#### 5.3.1.1 *Activation*

The activation step occurs when the user hovers or clicks the object of interest. The object of interest displays a visual feedback indicating that it can be dragged, and possibly a preview of the trajectory. As mentioned in [Dragicevic et al., 2011a], such preview is useful to show towards which direction the object can move. Using well established HCI mechanisms such as changing the arrow cursor to a double headed arrow may also be useful to indicate both the affordance and the direction. Once activated, the object of interest

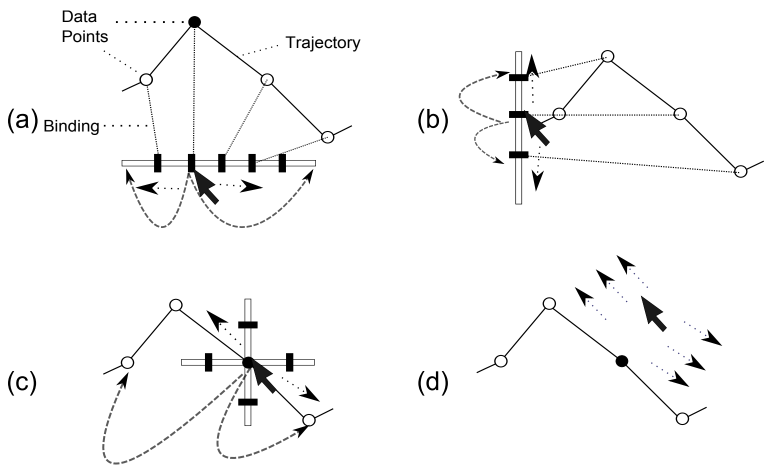



Figure 5.10: Dragging methods: (a) horizontal, similar as an embedded slider; (b) vertical, to explore values domain; (c) curvilinear, to follow loops; and (d) flow dragging to drag globally without having to select one object in particular.

is transformed into its trajectory. Previous research [Dragicevic et al., 2011b] shows that simple alpha blending can be visually sufficient, but staggered transitions may better communicate the trajectory's construction.

#### 5.3.1.2 *Dragging Methods*

While dragging an object along its trajectory may be intuitive, it is not always efficient. We explore alternative dragging methods. Each technique starts by directly dragging the object of interest, but then the user's mouse may shift outside the objects' perimeter to change values, while keeping the object activated.

*Horizontal dragging* is a 1-degree of freedom interaction, which maps the $x$ coordinate of the mouse to time, from left to right, as a slider does (Figure 5.10(a)). The mapping is an injective function, i. e. all the time points have a data value (assuming none is missing) thus all the data values are sequentially browsed in their temporal order. This dragging strategy is well suited to visualizations such as ranking tables, as the trajectories are injections.

*Vertical dragging* is also a 1-degree of freedom interaction, which maps the $y$ values of the mouse to domain values, from bottom to top. It makes snapping to minimum and maximum values easy by dragging downwards or upwards. However, it allows exploring all the values only if the trajectory is strictly monotonous (Figure 5.10(b)).

*Curvilinear Dragging* is a 2-degree of freedom interaction constraining the drag gesture along the trajectory. It enables both time and values exploration, and trajectory crossing ambiguities can be solved using a 3D distance [Dragicevic et al., 2008] taking into account the previous positions (Figure 5.10(c)).

*Flow Dragging* [Dragicevic et al., 2011a] is similar to the curvilinear dragging method, but applies to the background of the graphics (i. e. any area

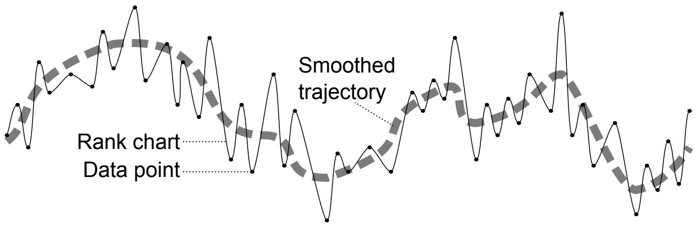



Figure 5.11: High variability can be addressed either by changing the height of the line chart, or simplifying the trajectory (displayed as dashed line, but usually not visible).

that is not an object of interest) to drag time rather than a specific object. This is particularly useful when the object is not visible yet, or when it is cluttered by too many elements (Figure 5.10(d)).

All dragging methods require additional parameters to become fully usable. As trajectories may be long, it is convenient to avoid scrolling the same distance as the object shifting. We introduce a *resolution* parameter which accelerates the mouse browsing of trajectories. From our observations, a good ratio is to scroll the width of the graphics to browse the complete time period. The the dragging methods also need to be *adaptive* to the topology of the curve, such as to smoothen high variability, while making sure the viewport remains centered. Both can be combined for global stabilization [Bederson and Boltman, 1999; Dragicevic et al., 2008] and can be calculated with a specific curve interpolation or curve fitting technique using line simplification.

#### 5.3.1.3 *Deactivation*

The deactivation phase occurs when the mouse drag is released. The trajectory is transformed back to the object of interest representation and the data graphic is updated to the selected time value.

One artifact of interpolating data value is that once the activation phase is over, the user may release the object of interest at a position that is not a data value. In other words, the animation is stopped before being terminated. Thus, snapping the animation to the nearest value is necessary. An alternative strategy consists of snapping to the latest browsed ranking.

### 5.3.2 *Examples*

To demonstrate both the trajectory design space and the genericness of the interaction state model, we explore three implemented examples: bar chart, scatter plot, and ranking table. Our prototypes are available online [5] and will be released as an open source project. We detail the main design decisions, particularly the identification of the objects of interest and the design of the trajectory. The most difficult decisions are related to the dragging method choice and parameters.

5 http://90plan.ovh.net/~lafrancep/timenav/

#### 5.3.2.1 *Bar Chart*

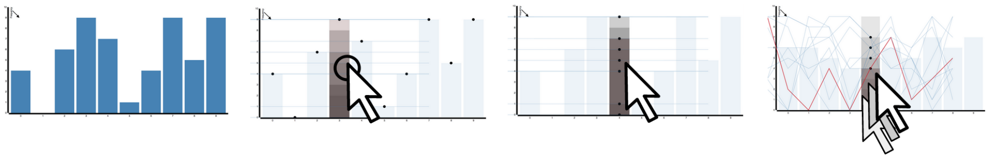


Figure 5.12: Illustration of time navigation with a bar chart. Objects of interest are the bars. The mouse pointer is displayed with a circle if a click is performed, and a series of arrows if a drag is performed.

A bar char encodes numerical values with the length of a bar, which can be vertical or horizontal. Multiple bars can be juxtaposed, often to display and compare categorical values. Updating the bar chart over time requires updating the height of each bar. Thus, each bar is considered as an object of interest, and their top border as the data point for trajectory, such as illustrated in Figure 5.12. In this Figure, the horizontal lines show the values of the other dimensions—the other bars. The gray area is an overlay on the bar showing the possible values of the object of interest over time. When a user drags a bar, a series of animated transitions transforms all bars—not only the bar of interest—into trajectories that are centered around the objects of interest. This way the variations of all bars are displayed. By showing trajectories over an existing chart, the x-scale becomes a time scale, as a replacement of the ordinal scale. However, as the line chart is visually different from the bar chart, we do not consider this as an issue. We implemented three dragging methods: vertical to drag by data values (which are visible as a gray overlay on the bar), and horizontal to drag by time. Because sometimes the bars are too small to be selected, we also implemented flow dragging, activated when dragging anything but the bars.

- Objects of interest: bars.
- Decorations: horizontal lines indicating other dimensions' values and a gray overlay area showing the possible values for the object of interest over time.
- Dragging method: vertical dragging (values), horizontal dragging (time), flow dragging (time).
- Transformation: none. Overlaid trajectory and data points.

#### 5.3.2.2 *Scatter plot*

A scatter plot displays the relationship between two variables. We used the GapMinder [Bostock, 2012; Rosling, 2009] dataset describing 180 countries over the last 209 years according to income and life expectancy. The objects of interest are the circles representing the countries. Because we found difficult to select a specific country, as some may be in dense areas, we used a Voronoi partitioning of countries to provide a perimeter of activation for every single country, for every time step. Figure 5.13 (center) shows the partitioning (which is normally not visible), making small countries easier to

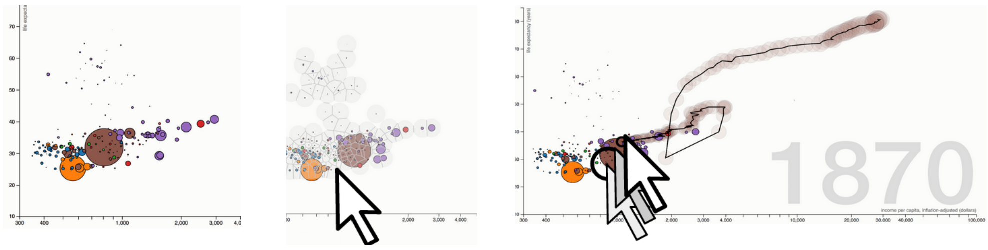



Figure 5.13: Illustration of time navigation with a scatter plot. Objects of interest are the circles representing countries. The mouse pointer is displayed with a circle if a click is performed, and a series of arrows if a drag is performed.

select. Nonetheless, dense areas require flow dragging navigation to change years, the background being the area outside the union of the objects' selection perimeters.

Conversely to previous examples, a scatterplot uses a 2D layout to position objects of interest in space, and objects of interest' motion trajectory is a 2D trajectory. Thus, time cannot be controlled using vertical or horizontal dragging. Instead, we used curvilinear dragging in 2D, involving to follow the path of the trajectory with the mouse cursor. While this strategy is more cognitively congruent, it requires steering thus may be more difficult to perform than horizontal and vertical dragging.

- objects of interest: circles.
- Decorations: overview of the data points position using transparent circles along the trajectory.
- Dragging method: curvilinear dragging (values) and flow dragging (time).
- Transformation: none. Overlaid trajectory.

#### 5.3.2.3 *Ranking Table*

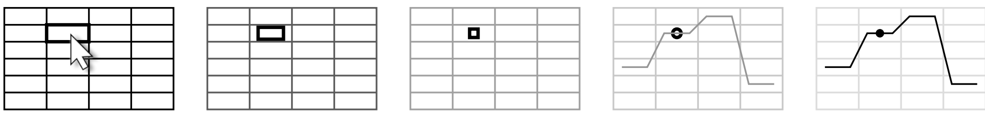


Figure 5.14: Transformation steps from a table cell to its trajectory.

We explored ranking tables design earlier. The two particular challenges this example raises are the unreachable objects of interest and the transformation stage .

Because objects of interest may be invisible on the viewport for tables with large numbers of rows, the interface scrolls down to follow the curve, allowing to reach all values while minimizing motion. Other entries' trajectories are displayed as background, as well as the original data table which is constantly updated based on the current position on the trajectory.

The table graphical elements are geometrically transformed to become data points on the trajectory. Figure 5.14 illustrates the strategy we used: from left to right, clicking a cell transforms its rectangular shape into a square, then into a circle, while the table fades out. The cell becomes the current data point on the line chart trajectory that fades in.

We implemented horizontal dragging for this example, because the ranking function is an injective function with no loop which can be controlled with one degree of freedom. Vertical dragging could also be used, but this strategy is problematic for non monotonous trajectories and introduces interaction ambiguity [Kondo and Collins, 2014]: when reaching an extremum, vertical dragging could mean going either right or left.

- Objects of interest: table cells.
- Decorations: data points representation, data causality, updated table in background, landmarks.
- Dragging method: horizontal dragging (time).
- Transformation: cell to data point, overlaid trajectory.

### 5.3.3 *Opportunities and Challenges*

Our previous case studies showed that with little technical and conceptual modifications, time-dependant standard graphics can benefit from trajectory-based interaction to better support time-related tasks. We now discuss the main opportunities and challenges raised.

#### 5.3.3.1 *Extension to Causality Analysis*

Data updates are usually directly connected to an event (e. g., new statistics on countries, soccer game results). We refer to the direct origin of the change as the *causality* of the update. Investigating data causality is important as graphics are sometimes updated without any change directly related to the dragged object.

For example, as detailed earlier, data causality can help soccer analysts strive to understand if the origin of the ranking's change is due to better performance of the team, or other teams that permute with it; and encoding the number of papers for the citations visualization helps understanding the cause of a researcher's progress. more generally, causality is valuable for *synoptic* tasks (which are about a set of values) because it better explains dependencies. Representing causality directly depends on the dataset and the task. While causality we implemented are static decorations, a promising challenge consists of making causality representations interactive. For example, when exploring ranking tables by horizontal dragging and highlighting the dependence between two objects of interest (Figure 5.8(b)), vertical dragging could be added to follow the dashed line linking the two objects of interest in order to navigate between objects of interest. Using the second spatial dimension of the mouse would then allow for switching from one object of interest, to the others, in a unique gesture.

#### 5.3.3.2 *Non-tangibles Variables and Mutable Scales*

Direct manipulation can only be performed on objects with coordinates. In some cases, such as with the bar chart, shapes with spatial boundaries may be dragged. Thus, non-spatial or *retinal* variables [Bertin, 1983], such as orientation, color, texture or symbols, need to be represented in a spatial domain to be manipulated (such as a color palette or a list of symbols). A naive solution would be to create as many additional views as necessary, breaking the initial assumptions of direct manipulation. Also, such variables may be encoded to depict existing objects with position. Another class of challenge is to deal with changing or *mutable* scales [Lumsdaine et al., 2012]. for example, such change would happen when new elements are dynamically added, or simply to accommodate the data being visualized (e. g., zoom in, zoom out).

#### 5.3.3.3 *Scalability and Targeting Objects*

The case studies we explored emphasized two aspects related to the difficulty of targeting objects (C6) and scalibility: targeting cluttered objects, and targeting objects beyond the viewport.

Time navigation in scatterplots raised the issue of targeting cluttered objects when they are in dense areas or when they are too small to be selected. To tackle this issue, we pre-computed a perimeter of activation for each object and at each time step, using a Voronoi partitioning of the 2D space. An alternative for targeting objects in dense areas would be to use excentric labeling [Fekete and Plaisant, 1999] to perform the selection.

By exploring time navigation in large ranking tables, we faced the problem of tables larger than the viewport. We proposed a way of identifying objects of interest beyond the viewport by keeping the activated object centered and translating the background visualization. As this strategy involves moving the background instead of the object of interest' representation, the mouse cursor does not the follow the object of interest anymore, introducing spatial indirectness. The simple way of avoiding the perception of this spatial offset we adopted is making the mouse cursor invisible during the drag gesture. Thus, it gives the illusion that the mouse cursor is the data point moving along the trajectory.

## 5.4 CONCLUSION

This chapter explored the direct manipulation of time-dependant data graphics using a transient trajectory interaction.

We first designed the technique for the particular case of ranking tables by displaying the trajectory of rows permutations as a ranking chart, and making them actionable to control animation. Then, we proposed the transient navigation technique as a general principle for direct and informative animated transitions in data graphics.

The prototypes we implemented and the exploration of several case studies showcased the rapid design, implementation and benefits of such a technique for time-related tasks, using diverse datasets. We explored the design variations and the interaction model for trajectory-based interaction, and finally discussed several promising directions for future work. The remaining of this section summarizes the empirical design guidelines this chapter leads to and analyzes the factors of directness of the technique.

### 5.4.1 *Design Guidelines*

Under the conditions that time is mapped to a spatial dimension and objects of interest can be selected, the following steps based on empirical observations provide guidance to implement such a technique:

1. *Objects of interest.* Identify the objects of interest, their dimensions, and time units over a time period. Objects of interest are activated and directly manipulated with physical actions. During the transition from the data graphic to the trajectory data points, it may be necessary geometrically transform the objects of interest representation (e. g., from a cell to a data point for ranking tables).
2. *Trajectories.* Design adequate trajectories, considering the various design variations we explored, to preview and follow the path of moving objects over time. A trajectory shows the path of the object and provides constraints to its manipulation. Hints (i. e. trajectory preview) provide guidance for activating the interaction technique to improve discoverability. Alternatives should be accessible to the user as the design variation choices depend on the current task.
3. *Dragging.* Decide on the appropriate dragging strategy (horizontal, vertical, curvilinear, flow). Loose dragging is tolerated between the trajectory and the user dragging location.
4. *Viewport.* Center the transition process around the focus point. Objects of interest should remain centered on the viewport during their activation. Non existent or invisible potential objects of interest should be made visible using viewport adaptation strategies.
5. *Graphical representations.* Complete with necessary decoration, landmarks, and any other graphic properties, by studying the task and the data domain. Trajectories may be decorated to show the values of a given object, over time, and may show data causality. Visual cues helping the dragging phase may appear only when the transient trajectory is activated.

### 5.4.2 *Directness and Benefits*

It is important to remind that time navigation in Infovis systems is usually performed using a dropdown list, or at best a slider, while advances in HCI make possible the design of more direct interaction techniques. This chapter is a step towards leaving behind the well-known but often inefficient widgets issued from the early ages of direct manipulation that dangerously hinder progress and popularity of information visualization. Here we refer to the summary of principles, benefits and challenges summarized in Table 2.1.

The generic technique we proposed here needs no widget at all (C15), has a high internal consistency (B13), and a high articulatory directness (P7). It result in a powerful yet simple interface, where a unique interaction allows for performing efficiently a large number of difficult tasks (C3).

The main limitation of Drag-Cell and Viz-Rank is their low temporal directness (P9, P12). Transient navigation using motion trajectories does not suffer from these issues. Indeed, the visualization is immediately updated as the user drags the object of interest along its trajectory such that the users can immediately see if their actions are furthering their goals and easily change direction (P1, P3, B5).

We also proposed a way of targeting objects of interest beyond the viewport by keeping the activated object centered and translating the background visualization (C6).

DEGREE OF INTEGRATION A good instrument should have a high degree of integration (P13), such as for the scatterplot where curvilinear dragging has a degree of integration of 2/2. However, it is sometimes preferable to limit the degree of integration to 1/2 for time navigation. For example, 1D horizontal dragging is well suited to ranking tables where trajectories are injections. Constraining the gesture to one degree of freedom avoids user's steering, a slow motor task. Similarly as horizontal dragging for ranking tables, a circular chart such as a pie chart would benefit from a constrained polar trajectory. The gesture remains free, but the cursor is snapped to the trajectory. When trajectory dragging only requires one degree of freedom, the second spatial dimension can be used for additional navigation strategies, such as switching from one object of interest to another in ranking tables.

▷ A high degree of integration is not always required, and the degrees of freedom of the input device can trigger different navigation modalities.

This technique tackles the difficult challenge of interacting with a three-dimensional information space using a two-dimensional input device (C16). This challenge is related to the congruence of the interaction technique.

OBJECT OF INTEREST AND CONGRUENCE The first implication for designing trajectory-based interaction consists of identifying the objects of interest. As soon as the object of interest is a graphical object, using a separate widget makes the interaction non congruent. Indeed, the semantic of the task rarely involves to jump at a specific time, as seen in Chapter 4. On the opposite, directly dragging the objects towards a direction makes these questions easy to answer and showing other objects of interest trajectories makes comparison straightforward. Conversely to Drag-Cell and Viz-Rank, it results in a high degree of compatibility (P14) and a high semantic directness both in the gulf of execution (P5) and in the gulf of evaluation (P6). Moreover, the technique we propose is based on physical actions (P2) performed directly on the objects of interest (P11); according to the dragging strategy, these physical actions have a medium to high degree of integration (P13). The first implication for designing congruent interaction resulting in a natural mapping to the task (P8) is then:

▷ Identifying the objects of interest is the first consideration to design cognitively congruent interaction.

SEAMLESS INTERACTION Another benefit of the technique is the seamless and playful interaction it provides. The technique has a high semantic directness in the gulf of evaluation (P6), giving the illusion to manipulate directly the semantic objects (B9). Because featuring a unique interaction seamlessly transitioning to the time domain (P15), no mode changes are required (P18). Finally, the absence of widgets by manipulating directly the graphical objects of interest makes the interface invisible to the user (P16).

The interface is also *playful* and *engaging*, reducing users' anxiety (B6) and making them eager to show the system to novices (B8): it is non interfering (P10), with no error message (B4), and interaction is rewarded using seamless transitions with smooth animations (P17). User's enjoyment was clearly confirmed when informally showing the prototypes to potential users, who could not stop 'playing' with the dataset as soon as the data was of interest for them. our interaction model expresses the fluidity of the interaction steps that are all integrated in a unique gesture that can be stopped at any time without consequences.

▷ Seamless and playful interactions that engage users and make them feel in control require forsaking standard widgets.

Finally, from on our own experience designing prototypes and from feedback we collected when showing these prototypes to potential users, we realized the importance of a visually appealing interface which is graphically customizable. This point is important as aesthetics in Infovis are usually considered as a second thought [Dix and Ellis, 1998] though an important factor for increasing Infovis techniques popularity.

▷ Interaction aesthetics are a factor of attractiveness and engagement and should not be an afterthought.

# 6 NEW INTERACTIONS FOR CRAFTING TABULAR VISUALIZATIONS

This chapter is based on previous publications [1], [5]. Thus any use of "we" in this chapter refers to Charles Perin, Pierre Dragicevic, and Jean-Daniel Fekete.

As noted by Dix and Ellis, *"Tables have been used since the earliest civilizations with examples of Mesopotamian tablets with tables of Cuneiform numbers dating back to 2900BC."* Despite the fact that tables are certainly the most common way of representing data, paradoxically there are very few novel interactions designed for tables. This chapter presents Bertifier, a web application for rapidly creating tabular visualizations from spreadsheets. It illustrates how a legacy visualization technique (tabular visualization) can benefit from interaction. Bertifier draws from Jacques Bertin's matrix analysis method, whose goal was to "simplify without destroying" by encoding cell values visually and grouping similar rows and columns. Bertifier remains faithful to Bertin's method while leveraging the power of today's interactive computers. Tables are formatted and manipulated through *crossets*, a new interaction technique for rapidly applying operations on rows and columns. We also introduce *visual reordering*, a human-steered reordering approach that lets users apply and tune automatic reordering algorithms in an interactive manner.

This chapter addresses the following questions:

▷ How to leverage advances in HCI for redesigning an analytic method based on physical manipulation?

▷ How to support human-steered reordering algorithms in a simple and visual way?

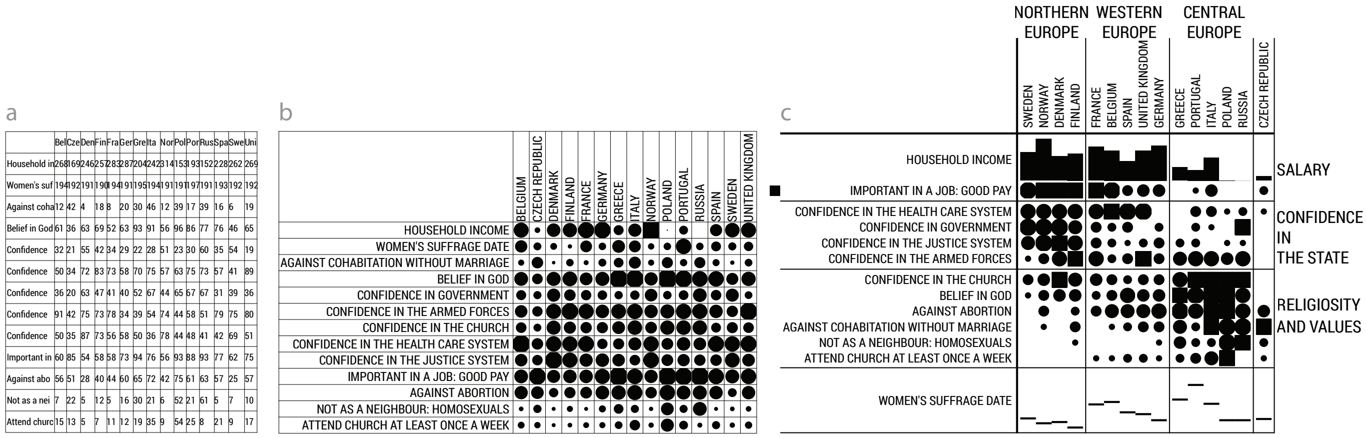


Figure 6.1: A spreadsheet formatted and reordered with Bertifier: a) the original numerical table; b) the corresponding tabular visualization; c) the final result, reordered, formatted and annotated. The final result is ready to be exported and inserted as a figure.

Most Infovis researchers know the French cartographer Jacques Bertin from his 1967 monograph *"La Sémiologie Graphique"* [Bertin, 1967]. Less known is his later work from 1975, *"La Graphique et le traitement graphique de l'information"* [Bertin, 1975, 1981]. This book details a method for processing and communicating tabular data visually that was meant to be effective, generic and accessible to any scientist and researcher [Palsky, 2003]. It was based on two simple ideas: *i)* encoding table cells visually, and *ii)* grouping similar rows and columns to reveal patterns. Bertin devised and refined his method after years of work with data analysts such as geographers, agricultural economists, ethnologists, and historians [Palsky, 2003]. However, his method required a physical matrix that was only available at his lab and involved tedious manipulations—often weeks of work.

Now with the widespread availability of computers, anyone can generate tabular visualizations automatically [Wilkinson and Friendly, 2009], and Bertin's physical matrix is little more than an intriguing historical anecdote. However, the results are rarely satisfactory without a good amount of user interaction [Henry Riche and Fekete, 2006; Siirtola, 1999]. Bertin himself realized the formidable potential of interactive computers [Palsky, 2003] and he tried to adapt his method to computers [Bertin and Chauchat, 1994]. Later, some Infovis researchers tried to resurrect Bertin's matrices [Sawitzki, 1996; Siirtola, 1999]. However, none of these implementations has been widely adopted, perhaps because they only featured rudimentary interactions, were incomplete, and were not accessible to a wide audience.

Bertin's idea to make tables visual and fully reorderable remains largely underexploited. Yet we believe that such an approach can facilitate data analysis not only by scientists, but also by a much wider audience due to the strong similarity between common tables and tabular visualizations. Many people today store and manipulate data using spreadsheets, and may be interested in getting new insights on their data, effectively communicating it to others, or quickly making sense of tables they receive. Although spreadsheet tools offer basic support for tabular visualization and reordering, through conditional formatting and sorting, this support remains surprisingly poor. We address this lack of proper tools through the following contributions:

- Requirements for tabular visualization tools based on Bertin's work.
- An extensive review of existing systems for tabular visualization.
- Bertifier (`www.bertifier.com`), a web-based tabular visualization authoring system compatible with online spreadsheets.
- *Crossets*, a novel interaction technique for creating, manipulating and fine-tuning tabular visualizations.
- Algorithm adaptations to support user-driven matrix ordering.
- A study suggesting that users from different backgrounds can use Bertifier to help them understand their own tabular data.

The tool we introduce remains faithful to the spirit of Bertin's original work while fully leveraging the power of modern graphical computers. By releasing it to the public, we hope that more scientists will consider Bertin's approach as a way to explore, analyze and interpret their data, as well as to communicate their findings by visual means. We also hope that Bertifier will bring Bertin's methods to a wider audience of both technical and non-technical users, and empower them with data analysis and communication tools that were so far only accessible to a handful of specialists, both for educational and for pragmatic purposes.

## 6.1 BACKGROUND

We first explain our focus on Jacques Bertin's method. We then present a short history of tabular visualizations with a focus on this method, and finally review existing implementations and the methodological issues involved in porting this method to graphical computers.

### 6.1.1 *Why Focus on Bertin's Method?*

As we discuss later in this Section, Jacques Bertin is not the only one who employed and advocated tabular visualizations and matrix reordering. Moreover, his recommendations are more informed by his intuitions as a cartographer than by formal empirical evidence. We however focus on Bertin for five major reasons:

- Jacques Bertin has accumulated a unique experience and know-how by working on real data analysis problems as a consultant for about 100 practitioners over the course of his career [Gardet and Irid, 2012; Palsky, 2003],
- Bertin's method is the only detailed tabular visualization analysis method that encompasses the whole analytic process, from formulating research questions to communicating insights.
- While Bertin's *Sémiologie* has had a great influence on the domain of Infovis, the full details of his matrix method are only known by a few. One of the goals of this work is to demystify Bertin's matrices and provide a tool with pedagogical and historical value.

- Sticking closely to Bertin's original work is a first step to understanding its limitations and identifying valuable improvements.
- Bertin introduced a unique visual style for presenting tables that people may want to imitate simply for its compelling value.

### 6.1.2 *Before Bertin*

Most of Bertin's ideas were not new. The idea of encoding cell values visually was already around in the late 19$^{th}$ century [Wilkinson, 1999]. We refer to this technique as *tabular visualizations*, but various other names have been employed, such as *heat maps* and *shaded matrix displays* [Wilkinson and Friendly, 2009]. The idea of reordering rows and columns to reveal patterns also dates back to the late 19$^{th}$ century, when the English Egyptologist Flinders Petrie reordered strips of paper to reconstruct the chronology of excavated graves [Liiv, 2010; Wilkinson and Friendly, 2009]. Although only rows were ordered and no visual encoding was used, tabular visualizations that were reordered on both rows and columns started to appear shortly afterwards [Wilkinson and Friendly, 2009].

Reordering tabular visualizations on both rows and columns manually is a tedious process, and specialized devices have been designed to assist in this task. In the 1950s, the Israeli mathematician and psychologist Louis Guttman built one called the "scalogram board" for analyzing questionnaire responses [Suchman, 1950]. The device consisted of two separate wooden boards, one for reordering rows and the other one for columns, and data items were represented by ball bearings that could be transferred between the two boards [Gibson, 1953]. In the 1960s French sociologist Robert Pagès built a magnetic version called "le permutateur" [Palsky, 2003]. Little information remains today about these two devices.

### 6.1.3 *Bertin's Method*

Despite this previous work, Bertin was the only one to provide a general and detailed method that was meant to be applicable across a range of domains [Palsky, 2003]. He also carefully considered visual design, building on his earlier work on map and chart design: *La Sémiologie Graphique* [Bertin, 1967].

Reorderable matrices are part of a more general method of "graphic information processing" [Bertin, 1969; Vergneault et al., 1967] that Bertin started to develop shortly after he published *La Sémiologie Graphique*. Later the method was detailed in a book on its own [Bertin, 1975, 1981] and summarized in new editions of *La Sémiologie Graphique* [Bertin, 1999; Palsky, 2003]. This method stems from his realization that, while maps and charts are traditionally drawn once for all, "mobile images" can be freely manipulated to reveal patterns [Bertin, 1975; Palsky, 2003]. Thus he recognized the importance of interactivity three decades before information visualization [Card et al., 1999; Dix and Ellis, 1998]. Bertin identified and physically implemented five types of interactive visualizations [Bertin, 1969]: the *family of*

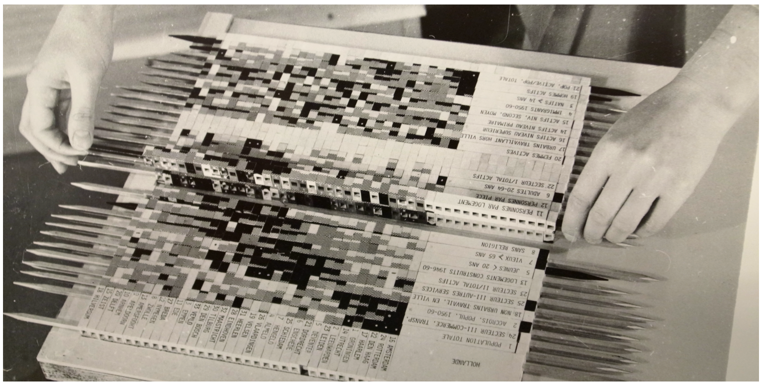


Figure 6.2: A physical matrix from Bertin (Serge Bonin, Archives Nationales).

*curves* (line charts on transparent paper sheets that can be superimposed), the *image file* (stacked cards with a visual index), the *collection of maps* (rearrangeable small multiples), the *collection of ordered tables* (rearrangeable paper tables), and finally the *reorderable matrix*.

The general method consists in three stages [Bertin, 1975, 1981]: **S1** – *Convert research questions into a table*, i.e., **S1a**: frame high-level research questions, then **S1b**: compile data into a numerical table in a way that is relevant to the research questions. Then **S2** – *Construct and process the image*, i.e., **S2a**: turn the table into an image by choosing the appropriate encodings and data conditioning (i.e., data transformations), then **S2b**: manipulate this image in order to simplify without destroying and reveal hidden patterns. Finally, **S3** – *Interpret, decide and communicate answers* consists in interpreting the resulting image according to externally available information and annotating it for publication.

These three stages were carried out with the help of one of Bertin's physical interfaces, depending on the type of dataset considered. While all interfaces complement each other, the reorderable matrix is, according to Bertin, the most general method of which others are only subcases. Thus, the matrix method is the focus of our work.

Bertin designed three reorderable physical matrices he called Dominos, each with a different size and visual encoding. Figure 6.2 shows Domino 2, a matrix of intermediary size. A rod mechanism allowed unlocking either rows or columns for reordering. The initial stage, S1, was carried out on paper. In stage S2a, values were converted into discrete steps on a paper table (7 to 11 different steps depending on the Domino version), then the physical matrix was assembled by choosing among a collection of physical cells that encode different ranges of values. In stage S2b, the matrix was reordered. Finally, in stage S3, meaningful groups were identified and named. The result was then photographied or photocopied, and the final image was used as a figure in the scientific publication, enriched with a legend and caption.

The reordering stage S2b could take weeks because of the manipulations involved and because no systematic procedure was known. Bertin's reordering method heavily relied on visual judgment. When asked how exactly he reordered his tables, he referred to the *"painter's eye"*[1]. He however tried to give heuristics and illustrated them with several examples. He recommended the following [Bertin, 1975]: *i)* choose a row with a particular aspect (e. g., high values) and move it to an extremity of the matrix. *ii)* Move similar rows close to this reference row, and opposite rows to the bottom. This will create two opposite groups, with a third group in the middle. *iii)* Do the same for columns. *iv)* Iterate.

From Bertin's original matrix method we derive requirements that a computer implementation should provide:

R1 allow the creation of a table from raw data
R2 perform data conditioning: scale, clamp range, discretize (step), and inverse rows/columns values so they become comparable
R3 select an encoding for cell values
R4 present the table visually
R5 reorder the rows/columns to group similar items together and move apart dissimilar ones
R6 group rows/columns that form meaningful chunks
R7 annotate the matrix (name groups)
R8 finalize the results for communication / publication.

### 6.1.4 *Automatic vs. Manual Approaches*

Many of the previous requirements could in principle be automated, which raises the difficult question of the level of interaction needed to perform an effective analysis and to produce an effective tabular visualization. Bertin's method generally involves many different types of tasks. Some of them have a strong data analysis component (e.g., stage S2b), while others clearly require a substantial amount of decision making. The stage S2a, for example, involves choosing appropriate visual encodings, a task for which computer automation can provide only limited assistance. The same is true for stages such S1a and S3.

What remains is the question as to whether the stage S2b (requirement R5) should be fully automated or whether humans need to be involved. For data analysis in general, the question of automation is a long-standing debate [Fekete et al., 2008]. While fully automated analysis yields tremendous benefits such as speed, scientific objectivity and reproducibility [Green Jr, 1966], full automation also has a number of issues: *i)* it ignores the analyst's knowledge that cannot be formalized [Wegner, 1997]; *ii)* it yields only minimal exposure to the data and therefore does not encourage deep understanding [Few, 2013]; *iii)* it may produce errors that can be way more costly than the effects of human subjectivity [Wilkinson, 1999]; and *iv)* it often produces visualizations that are suboptimal for communication. For example, cluster analysis throw out lots of data and can produce results that are controversial, subject to misinterpretation, or meaningless [Guttman, 1977; Liiv, 2010; Siirtola and Mäkinen, 2005].

1 J. Bertin, personal communication, 2008.

Many researchers have stressed the importance of human judgment in data analysis. Bertin was one of them: "the best graphic operations are those carried out by the decision-maker himself" [Bertin, 1981]. But Bertin was also aware that matrix reordering could benefit a lot from automation [Bertin, 1967], at no cost: automatic reordering is a non-destructive analysis approach, just as manual reordering. However, Bertin commented on the results of three automatic reordering algorithms and pointed out that none of them was satisfactory [Bertin, 1999]. He concluded that automatic reordering can save time but needs to be interlaced with manual tweaking. Semi-automatic approaches to matrix reordering have been argued for since then [Henry Riche and Fekete, 2006; Siirtola and Mäkinen, 2005]. Devising semi-automatic methods however requires a good understanding of automatic methods.

### 6.1.5 *Matrix Reordering Methods*

Automatic matrix reordering methods have been extensively studied, although mostly in isolation by three domains: statistics, linear algebra, and graph theory. Liiv [Liiv, 2010] provides an extensive review of seriation (another term for matrix reordering). According to him, "seriation is an exploratory combinatorial data analysis technique to reorder objects into a sequence along a one-dimensional continuum so that it best reveals regularity and patterning among the whole series." While clustering methods try to create cohesive groups according to some measure of cohesion, seriation is only concerned by finding an order, leaving operations like grouping to the interpretation of the user.

Automatic seriation of a table $T_{N,M}$ of $N$ rows and $M$ columns consists of finding an order for the rows and columns that optimizes and objective function $F(T)$. A naive method would try to evaluate $F$ for all the possible permutations of rows and columns, requiring $\frac{N!M!}{2}$ operations, which is not practical even for small tables.

Furthermore, even the objective function $F$ is not well understood. According to several articles, $F$ should bring together rows and columns that are similar. This characteristic has been formalized by Robinson [Robinson, 1951] by using the similarity matrix (*SM*) of rows (resp. columns), considered as vectors, computed using a similarity measure. According to Robinson, "the highest values in the matrix should be along the diagonal and monotonically decrease when moving away from the diagonal" [Robinson, 1951]. There are several automatic methods to compute an order to try to achieve a "Robinsonian" *SM* from a given *SM*, but for most real tables, no permutation will lead to a truly Robinsonian *SM*. Approximations of the Robinsonian can always be computed, but there is no clear metric to decide which is best.

Still, even without a clear characterization of "good" orders, there is a rich literature on automatic methods for seriation; Liiv cites 171 articles in his review [Liiv, 2010]. One approach is to model a numeric table as a weighted bipartite graph, the edges being the table cells connecting rows to columns, with the cell values as weights. Graph-based ordering methods can then be applied. Díaz et al. [Díaz et al., 2002] present an overview of linear ordering methods for graph vertices that try to optimize 9 graph measures (e. g., bandwidth or min-cut), referring to 261 articles.

Following Bertin, we want to let users interactively build a subjectively satisfying order by iteratively enhancing many partial solutions. Bertin argues for ordering rows and columns independently, which is incompatible with methods ordering both axes at the same time such as biclustering or Siirtola's approach [Siirtola, 1999]. We thus opted for using "optimal leaf ordering" [Bar-Joseph et al., 2001]. This method starts from a hierarchical clustering of rows (resp. columns) and finds an order that is consistent with the dendrogram and optimal in the traveling salesman's sense: the sum of distances between consecutive items is minimal. This method is computationally expensive ($O(n^3)$ in time) but provides very good results with only two parameters: the distance metric and the linkage type used for the hierarchical clustering. Furthermore, it is flexible enough to accommodate enhancements for interactive use.

### 6.1.6 *Modern Tabular Systems*

We now turn to existing systems. Before reviewing systems inspired by Bertin's reorderable matrices, we give a brief overview of other tools for creating and manipulating tabular visualizations.

Spreadsheet tools (e. g., Excel, Google Spreadsheet) are probably the most commonly used. Their primary purpose is to execute functions on cells or subsets of cells to compute derived values. They can easily be used to create tables from raw data (R1) and condition rows and columns (R2). As for R3 and R4, it is possible to use conditional formatting to encode cell values, typically through color — although Bertin strongly argues against the use of colors to encode numerical and ordinal values, since colors have no natural order [Bertin, 1975]. Spreadsheets also include various charting tools, but these are not tabular visualizations. Concerning reordering (R5), most spreadsheet tools support manual reordering by letting users drag row or column headers, and also support sorting, a useful but limited form of automatic reordering.

While spreadsheet tools are mainly designed to create and manipulate tables, other programs are specifically meant to visualize them. Tableau [Tableau Software, 2014] supports a large number of visual representations for tabular data, including tabular visualizations (R4) for which it provides several cell encodings such as color or bar charts (R3). It also supports data conditioning (R2), manual row and column reordering (R5), annotations (R7) and image export (R8). Thus, despite not being (at least explicitly) inspired from Bertin, Tableau supports a range of features recommended by Bertin. However, Tableau does not provide automatic reordering algorithms besides sorting (R5), and—like most Bertin implementations as we will see—it relies

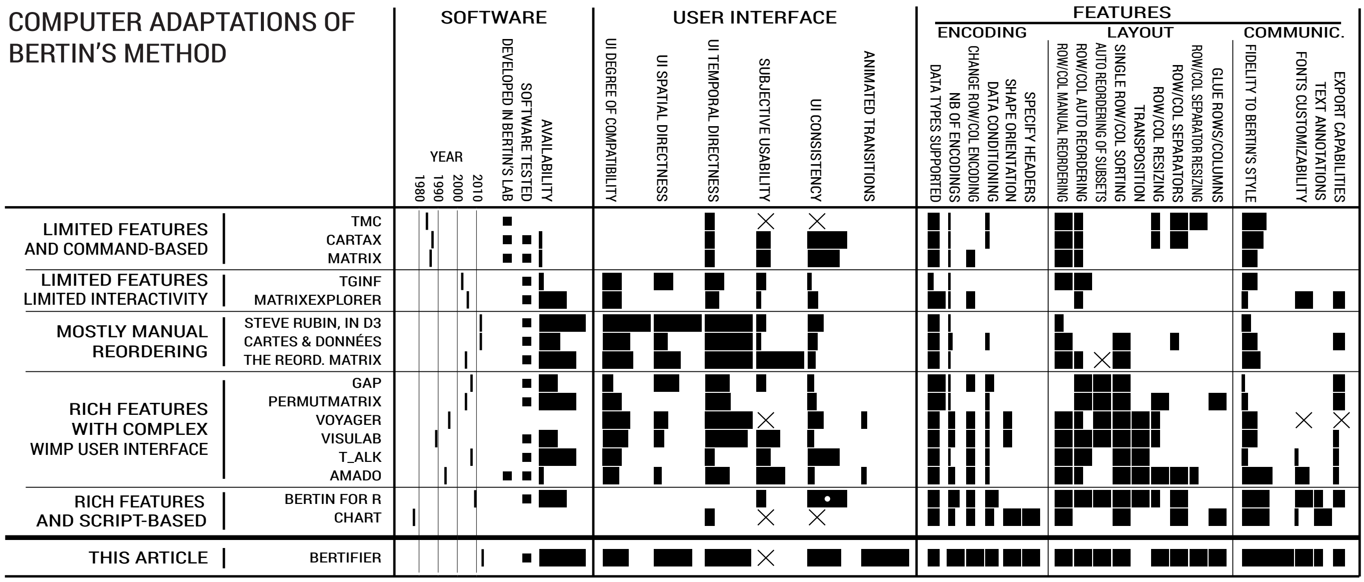



Figure 6.3: Computer adaptations of Bertin's matrix method with their availability, interactivity, and features. Crosses indicates unavailable data.

on a complex WIMP-style user interface. Several other charting and visualization systems exist that usually support image export (R8) but are often weak in interactive reordering, grouping, and annotation (R5–R7).

Several Infovis research systems implement or extend tabular visualizations. Table Lens [Rao and Card, 1994] is a focus+context system for exploring large tables, where numerical rows and columns can be interactively compressed into tabular visualizations. Users can change visual encodings (R3) and sort columns (R5). FOCUS [Spenke et al., 1996] is another focus+context tabular exploration system that lets users collapse identical adjacent values. Tableplot [Kwan et al., 2009] is a system for visualizing factor-analytic data that extends tabular visualizations with elaborate cell encodings, including multidimensional encodings (R3), and supports automatic reordering (R5). None of these tools has grouping, annotation, or export capabilities (R6–R8), making them inappropriate for communication.

### 6.1.7 *Computer adaptations of Bertin's method*

The number of past attempts to adapt Bertin's method to computers is surprisingly large. While previous work only cite a few of these attempts [Caraux and Pinloche, 2005; Henry Riche and Fekete, 2006; Hinterberger, 2010; Sawitzki, 1996; Siirtola, 1999], we conducted an extensive search over the course of three months and found a total of 16 tools, reported in Figure 6.3. These tools were built within different communities who were likely not aware of each other's work. Several tools were not available online and were obtained by contacting the authors or their past collaborators. The three tools that could not be tested at the time of this article's submission were evaluated based on publications, videos, and communication with their authors.

Figure 6.3 presents systems in rows and their characteristics in columns. The *features* category refines our requirements R2-R8 into testable software functionalities based on two extra sources: Bertin's later specifications for adapting his method to computers [Bertin, 1982], and our own observations of images created by him and his collaborators. The *interaction* category assesses the user interface. Although Bertin emphasized interaction, he did not discuss how the user interface should be designed. The tools listed employ a variety of interaction styles, ranging from fully conversational (e.g., command-based) to fully graphical (e.g., direct manipulation). Direct manipulation interfaces are generally believed to be easier to learn and to support rapid, incremental and reversible actions [Shneiderman, 1983]. Thus the first three columns score interactivity by how "direct" the interactions were, based on criteria from the instrumental interaction model [Beaudouin-Lafon, 2000]. More details on the coding schemes are available at `www.aviz.fr/bertifier`. Based on these scores we classify the systems into five categories:

1. *Limited features and command-based:* TMC [Le Fourn and Mailles, 1984], Cartax [Gimeno and coise Durand, 1988], and Matrix [Fras et al., 1984] are early systems developed in Bertin's laboratory and used in French public schools. They support only a few visual encodings (R3) and layout features (R4), and limited data conditioning (R2). They support both manual and automatic reordering (R5), but some of the developers commented that automatic reordering was acting as a black box and recommended using pen, paper, pen and tape as a first step to get used to the method [Girault and Rukingama, 1988]. From the interaction standpoint, the three systems are based on keyboard input and are clearly obsolete.

2. *Limited features and interactivity:* these systems support R3, R4 and R5 but are lacking most features and employ indirect interaction methods (i. e. menus, toolbars and dialog boxes). TGINF [Orús and Villarroya, 2003] provides few features and only accepts binary values. MatrixExplorer [Henry Riche and Fekete, 2006] has a few more features but at the expense of a more complex user interface.

3. *Mostly manual reordering:* a d3 implementation [Rubin], Cartes et Données [ARTICQUE, 2013], and The Reorderable Matrix [Siirtola, 1999; Siirtola and Mäkinen, 2005] are relatively recent implementations that support R3, R4 and R5, and let users rearrange the tabular visualization by direct manipulation. Their user interface scores high in overall "directness" but only because the system has very few features besides manual reordering.

*4. Rich features with complex WIMP user interface:* GAP [Wu et al., 2010], PermutMatrix [Caraux and Pinloche, 2005], Voyager [de Falguerolles et al., 1997; Sawitzki, 1996], VisuLab [Hinterberger, 2010], T_alK [Gosselin, 2008], and AMADO [Bertin and Chauchat, 1994] generally support most of the requirements: R1, R2 to some extent, R3, R4, R5, and for PermutMatrix [Caraux and Pinloche, 2005], R6. In contrast with previous systems, they provide table layout features and support the reordering of subsets, hence a better support for R4 and R5. They are still lacking in encoding flexibility (R3) and do not support text annotations (R7). They employ indirect interactions like the systems in category 2, but are generally high in temporal directness due to the tools having an immediate effect, and some use of direct manipulation.

*5. Rich features and script-based user interface:* Two tools use a script-based language for creating and customizing tabular visualizations. CHART [Benson and Kitous, 1977] is the first computer implementation of Bertin's method. The paper describes a long list of features but we could not test the software. Bertin-R is a plug-in to the R statistical package [Sawitzki, 2012] that provides powerful features and capitalizes on R's extensive support for data importation (R1) and transformation (R2). Both systems are close to supporting all features (R1-R7) but they employ a user interface that is indirect and dedicated to experts. The author of Bertin-R—also author of Voyager [Sawitzki, 1996] in Category 4—recognized that such interfaces are incompatible with Bertin's vision of "mobile images"[2].

To summarize, there were a remarkably large number of attempts to bring Bertin's method to computers over more than 30 years, but none of the implementations really caught on. We may have a working system today had developers built on each other's accumulated experience, but most of them were not aware of all other attempts. Our survey tries to fill this gap. Overall, some early attempts—especially CHART [Benson and Kitous, 1977] in 1977—were rich in terms of functionality, but poor in terms of interactivity. On the opposite side of the spectrum, a few recent implementations—Like Rubin's implementation in d3 [Rubin]—provide consistent, direct and easily-discoverable interactions, but at the cost of only providing a few features. Like in many other application domains, there is a marked trade-off between the level of functionality on one side, and the level of interactivity and discoverability on the other side. But the two sides are not entirely irreconcilable. The last row shows the features of BERTIFIER, the system we propose, and suggests that a Bertin implementation can be high overall both in functionality and in interactivity. We discuss this system in the next sections.

2 G. Sawitzki, personal communication, February 11, 2014.

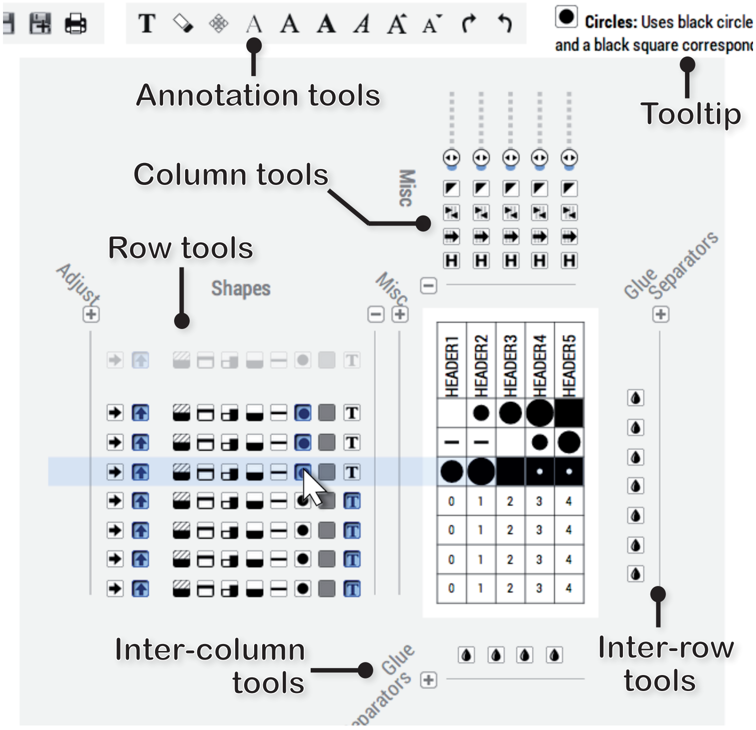



Figure 6.4: The BERTIFIER interface, with some tool groups collapsed.

## 6.2 BERTIFIER WALKTHROUGH

We illustrate BERTIFIER based on a fictional usage scenario. Suppose Clara is preparing a newspaper article on European values. She compiles a Google Spreadsheet where each column is a country, and each row contains information about this country. She took most of her data from a survey on European values [EVS, 2011] : for example, the percentage of people who believe in God, or who find that a good salary is important for a job. Although she carefully selected the data to keep the table small (15×16), it is impossible for her to see any pattern.

Clara prefers not to use traditional charts (e.g., bar graphs or scatterplots) because she wants to see the entire table. So she decides to "bertify" it. She just publishes her spreadsheet to the web, opens the BERTIFIER web page and pastes the table URL. She sees the numerical table (Figure 6.1(a)) and can transform it in a number of ways.

When Clara moves her mouse outside the table, a gray frame appears that contains formatting tools made of *crossets*: small crossing-based interactive components (Figure 6.4). All tools on the left apply to entire rows, while tools on the top apply to columns. Tools on the right apply to the space between rows, and tools on the bottom apply to the space between columns. When she moves her mouse on top of any tool icon, a tooltip appears to describe its function.

Tools are organized in groups. To mark the first row as a header row, Clara first expands the "Misc" group and clicks on the "H" icon adjacent to the first row (top right of Figure 6.5). She does the same with the left column. Then, to turn numeric values into shapes, she expands the "Shapes" group (second row of Figure 6.5), clicks on the circle icon next to the first row of the table to turn its values into black circles and squares. To propagate this encoding, she clicks the same icon on the next row, and drags down to the last row. This instantly turns all the table cells into circles and squares (Figure 6.1(b)).

Now Clara wants a more compact table. Each row can be resized independently using a slider placed next to it (leftmost crosset on the top of Figure 6.5): Moving the slider to the right increases row height while moving it to the left decreases it. She sets all rows to their minimum size by clicking on the topmost slider, dragging down until the last row (at which point all sliders are controlled concurrently), then dragging left. She does the same for columns.

Now she wants to tidy up the table. She drags vertically over all black arrow icons (top right of Figure 6.5), which reorders rows by visual similarity. Indicators that have a similar profile are now close together. She does the same for columns, which moves similar countries next to each other. To reduce clutter, she removes the grid by setting all white separators and black separators to their minimum value (bottom of Figure 6.5). Now, she can already see several country groups.

Clara continues to format the table to exhibit more patterns and convey a clearer message. This includes inverting the values of a row (importance of a good salary) to better show its correlation with another (household income), and emphasizing two rows of specific type (women's suffrage year and household income) by changing their visual encoding, increasing their height, and dragging them aside. Finally, Clara adds separators and annotations to emphasize groups. Her table is ready to be used as a figure (Figure 6.1(c)). She downloads it as a vector graphics file and inserts it in her word processor.

This figure will help Clara explain that there are roughly two opposite groups of countries: Northern Europe, with early women's suffrage, confidence in state organizations, happy with their high salaries, not very religious, and open-minded about homosexuality and abortion. On the opposite side, Central Europe countries trust the army and the Church but not their state organizations and justice system. Intriguing cases include the Russians who are very religious but do not go to church. Countries from Western Europe stand in-between. Czech Republic is an outlier: low income and low confidence in state organizations, but not religious. Clara will also comment on general trends. For example, salary tends to be more important in countries with low income. These countries also tend to be more religious and less open-minded. The women's suffrage came also rather late.


Figure 6.5: All crossets for manipulating rows and columns.

## 6.3 DESIGN DETAILS AND RATIONALE

BERTIFIER is an open-source application implemented using *Javascript* and *d3*, and runs in a modern web browser. Here we describe and justify the key aspects of its design.

### 6.3.1 *General Design Principles*

BERTIFIER's design philosophy is to remain as faithful as possible to Bertin's original method—in particular, we made sure we could reproduce most of Bertin's examples—while exploiting the opportunities of modern technology. Bertin's recommendations are influenced by the technology available at the time: very few things were possible or even thinkable, and HCI has improved dramatically since then.

As seen previously, Bertin's method is a step-by-step approach. Since the initial stage S1—formulating research questions and compiling a data table—is already well-supported by spreadsheet tools, we focus on supporting stages S2 and S3. In principle, data conditioning and visual encodings should be chosen first, then the matrix is built, then reordered, and then annotated [Bertin, 1975]. This sequential approach may have been driven by Bertin's devices, that did not allow otherwise. We take a different angle by making all these operations accessible at any time and from a single integrated view, without enforcing any order. Thus BERTIFIER is compatible with Bertin's step-wise approach, while supporting more flexible data exploration and visual authoring processes by minimizing premature user commitment [Green and Petre, 1996].

Another design principle is the accessibility to a wide audience. This includes the choice of a stand-alone web application, support for online spreadsheet import, the design of a short video tutorial, the use of informative tooltips, and a careful choice of terminology. For example, a term like "visual encodings" is unlikely to be understood by people without Infovis background, and we therefore preferred the term "shapes" (Figure 6.5). Animated transitions are also provided for all operations to help users understand their effects.

Two major design problems remained to be addressed: designing tools that are simple yet powerful enough to support a wide range of operations on matrices and subsets of matrices, and providing a sensible support for human-assisted reordering. We addressed the first issue by introducing *crossets*, and the second issue by introducing the principle of *automatic visual reordering*, presented in the next sections.

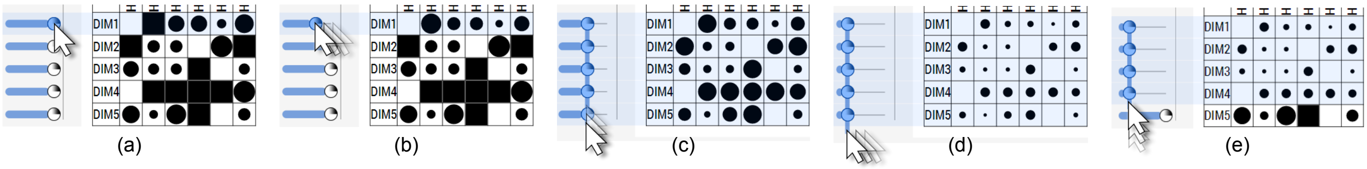



Figure 6.6: Crossing several crossets based on the continuous slider.

### 6.3.2 *Crossets*

Based on recent work in HCI, we chose to design tools that minimize the indirections typically found in traditional interfaces [Beaudouin-Lafon, 2000] and come as close as possible to direct manipulation. While manual reordering lends itself well to direct manipulation, it is not the case for more abstract operations such as choosing an encoding, or conditioning the values of a row. We addressed this problem by introducing *crossets*. Crossets are inspired from work on painting-based and crossing-based interfaces [Accot and Zhai, 2002; Apitz and Guimbretière, 2004; Baudisch, 1998], i.e., interfaces that let users invoke commands by crossing (or painting over) widgets instead of clicking them.

#### 6.3.2.1 *Two-dimensional Widgets*

Standard widgets have a spatial degree of integration of 0/2 (e. g., buttons) or 1/2 (e. g., sliders). Previous researchers have used the two spatial dimensions to interact with sliders. With the Infovis toolkit [Fekete, 2004], one can create multi-resolution sliders whose precision depends on the orthogonal distance to the slider. Orthozoom [Appert and Fekete, 2006] extends the scrollbar by integrating in the same instrument both the pan and the zoom: moving the thumb along the slider axis scrolls the page up and down, and the orthogonal dimension is used to zoom in and out the page. FaST Sliders [McGuffin et al., 2002] use the orthogonal dimension of sliders to make an additional menu pop up for further adjustments.

In Bertifier, the layout of the table is propitious to a grid layout where widgets can be aligned in front of the rows and columns. Thus, the orthogonal dimension of the widget can be exploited to manipulate series of widgets simultaneously. Moreover, the most efficient crossing gesture is *continuous* and *orthogonal* to the target [Accot and Zhai, 2002].

#### 6.3.2.2 *Design Space*

Crossets are based on the standard slider widget, enhanced with orthogonal crossing capabilities. Figure 6.6 illustrates the technique with continuous sliders: (a) selecting a slider's thumb activates the crossing gesture; (b) moving the mouse cursor horizontally allows for navigating in the values domain (here the visual encoding of the cells is updated); (c) moving the mouse cursor vertically allows for navigating in the crossets selection domain: crossed widgets are attributed the value corresponding to the $x$ coordinate of the cursor; (d) moving the mouse horizontally changes the value of all crossed widgets; and (e) uncrossing a widget restores its value before selection. Manipulating a crosset is performed in three steps:

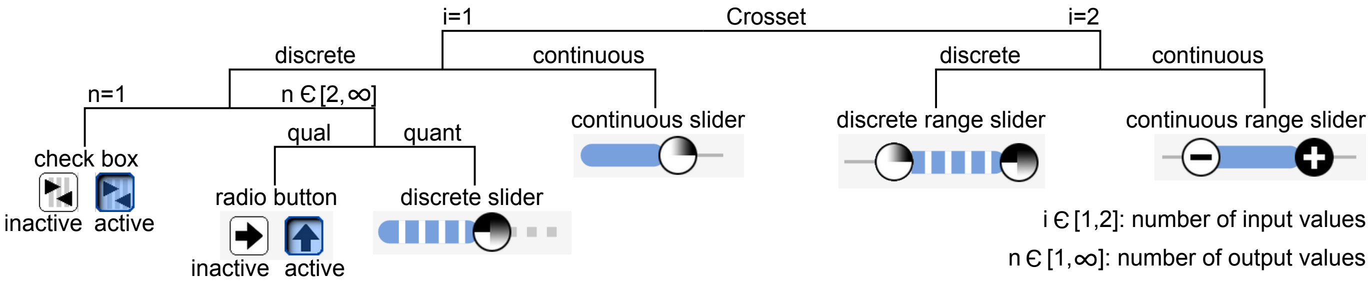



Figure 6.7: Crossets design space and implementation.

1. **Start:** the crossing starts when pressing the mouse button on the thumb/button of a crosset. The thumb/button visual appearance is updated to show its selection.
2. **Gesture:** moving the mouse in the two-dimensional space allows for navigating both in the values domain and in the crossets selection domain. The crosset graphical representation is immediately updated, and the orthogonal trajectory is displayed linking the first selected crosset to the mouse cursor to show the current crossing result.
3. **End:** the crossing gesture terminates when releasing the mouse button.

The design of crossets is based on the standard slider widget from which a wide range of widgets can be augmented with crossing capabilities. Figure 6.7 illustrates the design space of crossets that we implemented according to two parameters. $i$ is the number of variables of the crosset. In most of the cases, $i \in [1, 2]$, but sometimes $i > 2$. For example, Photoshop features 3-thumbs sliders to define color gradients. $n \in [1, \infty]$ is the number of accessible values (values domain of the crosset).

- The basic crosset is similar to the continuous slider ($i = 1, n > 1$).
- The discrete slider is a particular case of the continuous slider where reachable values are magnetized ($i = 1, n > 1$). The crosset visual representation emphasizes these discrete values.
- The continuous range slider ($i = 2, n > 1$) is a continuous slider with two thumbs to specify the min and max values. The implementation requires specifying a minimum gap between the min and max values to ensure thumbs selection and $min < max$.
- The discrete range slider is a particular case of the continuous range slider ($i = 2, n > 1$), like the discrete slider is a particular case of the continuous slider. The same implementation considerations apply.
- The radio button is similar to the discrete slider ($i = 1, n > 1$) but is dedicated to non-ordinal (qualitative) values. Each button can be seen as a step of a discrete slider and the selection is mutually exclusive (as illustrated in Figure 6.8) The visual representation of this crosset does not show the crosset axis as opposed to slider-based crossets, and the buttons have two states to identify which one is currently selected.
- The checkbox button ($i = 1, n = 1$) has two states: active and inactive. Because $n = 1$, it does not allow navigating in the values domain.

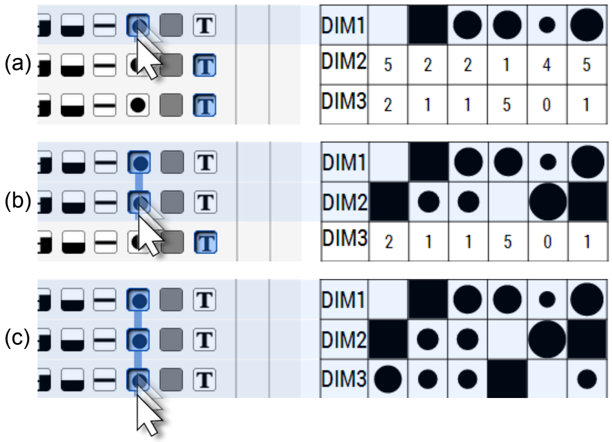



Figure 6.8: Crossing several crossets based on the radio button, with mutually exclusive selection. Here, the rows encoding is progressively set from text to circles and squares.

#### 6.3.2.3 *Crosset properties*

Careful design decisions were made such that crossets respect many principles from direct manipulation and instrumental interaction.

INTERNAL CONSISTENCY The crosset is a generic concept that applies to a wide range of standard widgets. Thus, it ensures the internal consistency of the interface, in terms of interaction, design and graphical properties. It favors initial learning (B12), ease of use (B13), and perceived quality (B14).

EXTERNAL CONSISTENCY Crossets have a high external consistency because they are enhanced standard widgets, featuring standard interactions. In particular, crossets are compatible with the click, consisting of the **Start** followed immediately by the **End** without considering the **Gesture**. The crossing gesture is triggered only when the user deviates its trajectory orthogonal to the widget axis. It favors fast learning and transfer of knowledge from the user (B15).

DEGREE OF INTEGRATION Conversely to standard widgets which have a spatial degree of integration of 0/2 (e. g., buttons) or 1/2 (e. g., sliders), crossets have a degree of integration of 2/2 because the two spatial input dimensions of the mouse are exploited (P13).

IMMEDIATE FEEDBACK AND REVERSIBLE ACTIONS The crossets graphical representation is continuously updated to provide visual feedbacks to the user (P1, P9, P6). Moreover, the result is immediate and reversible (P3), as illustrated in Figure 6.6(d,e).

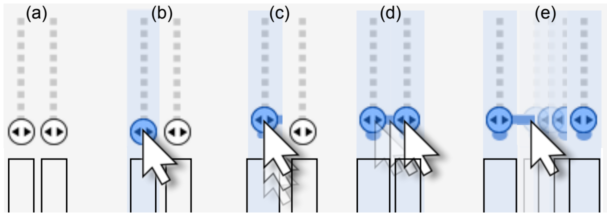



Figure 6.9: Issues occur when modifying the value of a crosset affects the position of one or several crossets.

TARGETING SEVERAL OBJECTS By navigating in the crossets selection domain, the gesture allows for applying the same action to a series of objects, one of the difficult challenges for direct manipulation interfaces (C8).

TEMPORAL DIRECTNESS Both the selection of objects of interest and the manipulation of values are performed in a unique gesture with immediate feedback. Thus, crossets minimize the temporal indirectness (P12) and the articulatory distance (P7).

CONSTRAINED GESTURE Because crossing is performed in only one dimension of the 2D-space, the gesture is free but its interpretation by the system is locked in the direction orthogonal to the crosset axis. This property ensures no error of selection [Accot and Zhai, 2002], without the need of intrusive error messages interfering with the task (B4, P10).

#### 6.3.2.4 *Layout Modifications*

When the action associated to a crosset affects one or several crossets' position, it may be impossible to immediately apply the result of the action. Let's consider the columns resizing scenario illustrated in Figure 6.9: modifying the value of the first crosset is not problematic (b,c); but crossing the second crosset moves it to the right and unselects it immediately because it is attached to the position of its corresponding column (d,e). The crosset value is then restored to its value before selection, and the crosset moves back to its initial position... to be instantaneously crossed again, because under the mouse pointer. As a result, the crosset and its associated column are undesirably translated from left to right, indefinitely.

Therefore, an alternate strategy is required for crossets whose action affect the position of other crossets. We adopted the same approach as the one implemented in window dragging in operating systems when the computer performances are low (only the frame of the window is updated while dragging and the content is updated once the mouse button is released). Similarly, we opted for the strategy of providing an incomplete preview of the result in real time and applying the layout modifications once the crossing gesture is ended (mouse release). It ensures crossets compatibility with actions affecting the layout, at the cost of introducing some temporal indirectness (P12).

#### 6.3.2.5 *Crossets for Bertifier*

In BERTIFIER, crossets are placed next to a row, a column, or next to an intersection (Figure 6.4). Crossets are grouped in "toolbars" that can be collapsed and expanded, and there can be as many crossets as needed. This allows users to apply a variety of operations on arbitrary rows or columns (e.g., for specifying headers or visual encodings), and on the intervening space between rows and columns (e.g., for specifying separators). The customization of individual rows and columns provides lots of flexibility. For example, Bertin produced tabular visualizations with different encodings for different rows [Vergneault et al., 1967]. Although these were made with another device than the reorderable matrix (the image file), being able to encode different types of dimensions in different ways can facilitate both data analysis and communication.

Crossets also allow users to apply operations to multiple rows or columns in a single gesture. For commands such as provided by the "Shapes" toolbar (Figure 6.5), this is done by applying the command to the first row, then crossing subsequent rows. Slider widgets (Figure 6.5) follow the same principle, except they also support the adjustment of multiple rows or columns at once, by dragging in the orthogonal direction. Compared to standard spreadsheet interactions for resizing rows or columns, the resizing crosset does not require the selection to be specified in advance.

By supporting crossing-based interactions, BERTIFIER makes it possible to quickly change arbitrary groups of adjacent rows and columns, which is useful in many cases, including for automatically reordering rows or columns within identified groups. Crossing all rows or columns allows users to apply operations to the entire table in a quasi-instantaneous manner, provided the table fits the screen. All these interactions remain consistent across all tools.

#### 6.3.2.6 *Actions affecting the table layout*

As introduced in Section 6.3.2.4, actions affecting the table layout require an alternative strategy. In Bertifier, three actions affect the table layout thus feature a partial feedback during the **Gesture** phase. The final result is applied only once the mouse button is released (crossing gesture **End**).

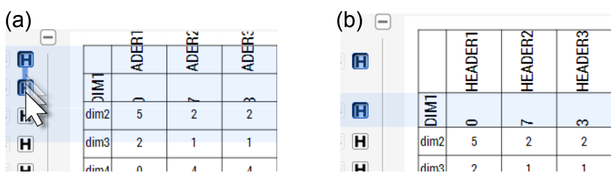



Figure 6.10: Preview as immediate feedback for text encoding.

Encoding the cells of a row or column as text affects the cells width or height, respectively. The partial visual feedback may include (Figure 6.10(a)) a rotation of the text, writing the text lowercase or uppercase, and changing the text anchor. Because the rows and columns are not immediately transformed, the text may temporary go over the edges of a cell. Once the crossing gesture ends, the rows and columns are smoothly animated to their new position or size (Figure 6.10(b)).

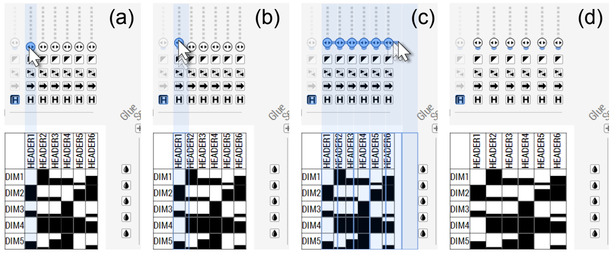


Figure 6.11: Preview as immediate feedback for column resizing.

Resizing rows and columns affects their height and width, respectively, and translates other rows and columns. Figure 6.11 illustrates columns resizing: after activating a crosset (a), vertically dragging its thumb affects the width of the associated column. The strategy consists of displaying the frame of the selected columns as they would behave if the drag was released as preview, in order to ensure that other crossable widgets remain still (b,c). Once the gesture is over, columns are smoothly animated to their new position and size (d).

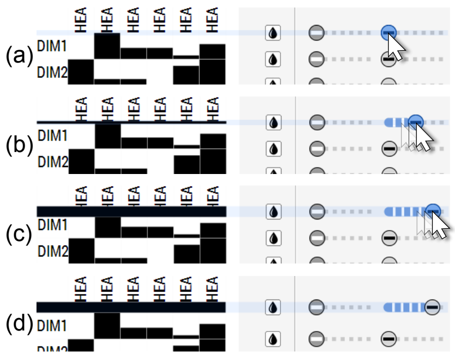


Figure 6.12: Preview as immediate feedback for separator resizing.

Finally, changing the values of separators affects the position of lines or columns as well as other separators. As illustrated Figure 6.12, (a) modifying the value of a discrete slider provides a preview of the currently modified separator. Because the rows and columns are not immediately translated to their new position while dragging the mouse, the modified separators are displayed above them (b-c). As previously, the affected rows, columns and separators are animated to their new position once the gesture is over (d).

### 6.3.3 *Human-Assisted Reordering*

Human-Assisted reordering is supported both through manual reordering interactions and through automatic visual reordering.

#### 6.3.3.1 *Manual Reordering Interactions*

As in previous implementations [Bertin and Chauchat, 1994; Rubin; Sawitzki, 1996; Siirtola, 1999], we support column and row reordering by drag and drop. This is the first level of integration between automatic and manual reordering, and allows to tweak the results of automatic reordering [Henry Riche and Fekete, 2006]. Following previous findings that reordering rows and columns concurrently can be confusing to users [Siirtola, 1999], we lock the reordering on the row or column based on the initial dragging direction. To help users understand changes, we perform animated transitions during the dragging operation.

Following previous recommendations [Siirtola, 1999, 2004; Siirtola and Mäkinen, 2005], we also support dragging on sets of rows and columns. We provide a tool that lets users "glue" several rows or columns together (Figure 6.5, bottom right). Since we distinguish between groups used for concurrent manipulation and visual groups (i.e., separators), we avoided the ambiguous term "group". Our automatic reordering algorithm preserves rows and columns glued together by the user, thus providing a second level of integration.

#### 6.3.3.2 *Automatic Visual Reordering*

The principle of *automatic visual reodering* is that rows and columns are reordered not according to the underlying data, but according to their visual similarity. We believe this principle is easier to understand for users not familiar with data analysis. Visual reordering is only a user interface metaphor that does not have to accurately capture what the system is doing, but which is meant to elicit a simple and "good enough" mental model of the system to allow for easy tuning.

The implementation of this metaphor is fairly simple and relies on two basic principles: *i)* the reordering algorithm should take as input the data *after* it has been conditioned and normalized (e.g., betwewn 0 and 1), and *ii)* the visual encodings used should ensure that visual differences are roughly proportional to numerical differences. From this it follows that the automatic reordering algorithm will behave *as if* it were operating visually. Next we discuss to what extent the condition *ii)* is fulfilled by the visual encodings implemented in Bertifier.

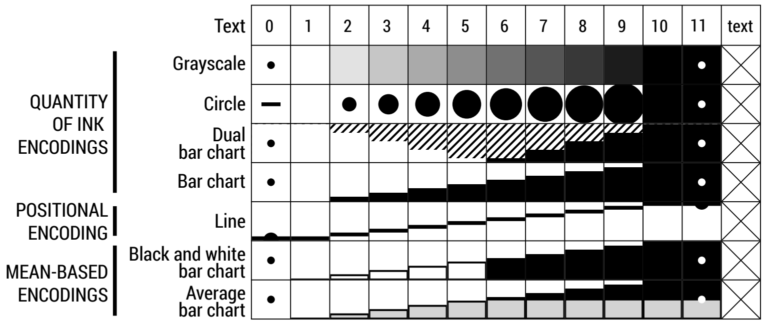



Figure 6.13: Encodings in Bertifier. The range of each row is set to [1, 10]. 0 and 11 are beyond the range. Crosses encode N/A values.

#### 6.3.3.3 *Visual Encodings*

Bertifier implements eight types of visual encodings (Figure 6.13). For the sake of generality, we consider text as being one type of encoding. Automatic reordering is however disabled on text to reinforce the visual reordering metaphor. All encodings except grayscale have been either explicitly mentioned in Bertin's book [Bertin, 1975] or were used by him. The software can be easily extended with other types of encodings.

The black and white bar charts and average bar charts are particular in that they encode summary statistics in addition to value: values lower than the mean are shown in white or gray, while values above the mean are shown in black. The line encoding is also different in that it uses a positional encoding. These three encodings have been only used occasionally and are not implemented in any of the physical matrices.

The remaining encodings are more common and all roughly follow the rule that the quantity of ink—the average pixel darkness—is proportional to the normalized data value. We enforced this rule to make the encodings consistent with the visual reordering metaphor. For example, the dual bar chart encodes the values from white to black, with a fully hatched cell corresponding to the mean value of the row. In this case, we use a 50% hatching to respect the rule of proportionality. Thus for these four encodings, computing a numerical difference between cells is equivalent to computing the difference in their average shading. The rule of proportionality is challenging to implement for the "circle" encoding, since the circle is clipped and progressively turns into a square. We chose to replicate Bertin's original encoding although previous work described an optimal scale for symbol size discrimination [Li et al., 2010]. Deriving the correct circle radius is a geometrical problem without any analytic solution, but we found the following good approximation:

$$r = \begin{cases} D = 2\sqrt{\nu/\pi} & \text{if } \nu \leqslant \frac{\pi}{4} \\ \frac{1}{2}(t + t^6)(\sqrt{2} - 1) + 1 \quad \text{with } t = \frac{D-1}{2/\pi - 1} & \text{if } \nu > \frac{\pi}{4} \end{cases}$$

with $\nu$ being the cell value and $r$ the (unclipped) circle radius.

#### 6.3.3.4 *Interactive Data Conditioning*

Although users can pre-process their data in their spreadsheet as part of step S1, BERTIFIER provides tools for performing further data conditioning of rows on-the-fly (the "Adjust" toolbar in Figure 6.5). Slider crossets are provided for *i)* adjusting the data range and clipping the values accordingly, *ii)* reducing the maximum value of the data (which has the effect of making all cells look brighter or disappear), *iii)* turning values into discrete steps. In addition, users can *iv)* invert row or columns values. *i)*, *iii)* and *iv)* have been recommended by Bertin [Bertin, 1982] and *i)*, *ii)* and *iii)* are implemented in CHART [Benson and Kitous, 1977].

These operations change the matrix visually in a similar way as photo retouching tools. Since the automatic reordering is performed on post-processed values, these controls provide a way of fine-tuning the reordering algorithm without explicitly tuning any of its internal parameters. Specifically, rows that are made brighter will be given a lower weight by the reordering algorithm. Rows that are made entirely white will be ignored. A specific case of this is making all rows white except one, which enables a regular sorting operation.

Sometimes it is useful to provide custom ranges (e. g., for applying a uniform range to all rows). To achieve this, users can specify the range of each row in the initial spreadsheet, using special header names. A crosset in BERTIFIER allows to enable or disable this custom range.

### 6.3.4 *Bertifier's Reordering Algorithm*

As explained in Section 6.1.5, we use the optimal leaf ordering [Bar-Joseph et al., 2001] as basis for visual reordering. This algorithm takes a list of vectors of values as input, either rows or columns, with a distance metric—we use Euclidean or Manhattan—and returns an optimal order by minimizing the sum of distances between consecutive vectors. We provide a richer API on top of the optimal leaf ordering to take into account interactions. The API allows to specify a first and/or last vector as limits that will remain at the end(s) and the remaining vectors will be optimally ordered. Finally, one or several ranges of vectors can be protected against reordering (rows or columns that are glued). Our implementation uses 4 pre-processing steps before the standard optimal leaf ordering: 1) limits management, 2) exception managements, 3) equivalent vectors management, 4) standard reordering.

Limits are implemented by changing the distance to the first and/or last vectors. These vectors should be far away from the other vectors so that the hierarchical clustering will add them to the last cluster(s). We compute the maximum distance $m$ in the matrix and add it to the distance of every vector to the starting vector. If there is an ending vector, we add $2 \times m$ to its distance to every other vector. Exceptions are implemented similarly: when a range $[a, b]$ is specified as an exception, all the vectors between them are removed from the list, only the first and last vectors are kept, and a distance of 0 is set between them in the distance matrix. The optimal leaf ordering then always glues them together and the indices of the vectors in-between are re-inserted afterwards.

The clustering algorithms can generate artifacts when applied with many identical vectors so we remove all but one of them before applying the algorithm and re-insert them afterwards after their representative. Note that the reordering algorithm can sometimes invert the order of vectors. When limits are specified, the pre-processing step 1) makes sure the final order keeps the limits where they were, but protected ranges can still be inverted by the algorithm. We believe this is not an issue for interactive reordering since range constraints are still honored and each group remains the same. Finally, standard sorting is just a particular case of the reordering algorithm when the list of rows contain exactly one column, and vice versa.

### 6.3.5 *Support for Requirements and Limitations*

Now we discuss to what extent Bertifier supports the requirements identified in the Background Section:

- R1 Table creation is delegated to spreadsheet software, whose data can then be imported in Bertifier. However, modifying the table requires restarting the whole authoring process. Also, the table needs to be correctly prepared, as Bertifier has no transposition feature.
- R2 Basic conditioning (scaling, clamping range, steps and inversion) is supported by crossets, while more elaborate processing needs to be done in R1. Adding more transformations (e.g., log) to Bertifier would allow to try more options with immediate visual feedback.
- R3 Encoding is supported by eight different crossets, plus two crossets to change their orientation (Figure 6.5). Encodings support ordinal data provided it has been numerically coded, while there is no encoding yet for qualitative data—which can be split into binary dimensions [Bertin, 1975].
- R4 The tabular visualization is always visible and updated on-the-fly.
- R5 Manual reordering is supported through direct manipulation and the glue crossets. Automatic reordering is supported through crossets, and can be tuned using the glue and data conditioning crossets. Implementing visual reordering based on actual visual similarity between cells (instead of ink) may yield new possibilities for algorithm tuning.
- R6 Crossets can be used to create a variety of separators. However, these are fixed and not updated when reordering rows or columns.
- R7 A separate toolbar allows users to add text annotations, but they are not structured and remain fixed when the table layout changes.
- R8 The results can be exported in SVG for further tweaking with external authoring software. This step is often crucial, for example for resizing labels or adding legends as in Figures 6.3 and 6.14.

## 6.4 ARE SPREADSHEETS COMPATIBLE WITH BERTIFIER?

To better understand spreadsheets generated "in the wild" we collected personal and professional spreadsheets by asking people from research (*C1*, 123 spreadsheets), and administration & education (*C2*, 128 spreadsheets) to send us their spreadsheets, for a total of 14 people. We extracted 16 characteristics per table. We use the notation $Q_{Ci} = \{Q1, Q2, Q3\}$ to report the quartiles of counts (25%, 50%, and 75%) of spreadsheets belonging to category *Ci*.

*Spreadsheets are generally compatible with Bertin's method.* Following Bertin, a table must clearly distinguish between entries and dimensions. 92.5% of our spreadsheets have explicit entries and dimensions for *C1*, and 93.6% for *C2*. Moreover, spreadsheets must contain numerical values and present a data table. 87% of *C1* spreadsheets meet this condition, and 85.2% for *C2*. The reasons for not being conform are various and include: containing multiple tables (44% of non conform spreadsheets for *C1* and 16% for *C2*); being a calendar layout (13% and 37%); being a simple list of people (0% and 26%); and being a drawing support or diagram layout (12% and 0%).

*Spreadsheets are small.* For numbers of entries, $Q_{C1} = \{13, 21, 37\}$: 25% of spreadsheets in *C1* are shorter than 13 columns, 50% are shorter than 21 columns, and 75% are shorter than 37 columns; $Q_{C2} = \{10, 22, 35\}$. For dimensions, $Q_{C1} = \{6, 9, 14\}$, $Q_{C2} = \{9, 11, 13\}$.

*Spreadsheets mostly contain quantitative values.* 64.4% of the dimensions are quantitative for *C1* and 92% for *C2*; then qualitative (14.9% and 5.3%), ordinal (10.2% and 2.5%), and text (10.5% and 0.2%).

*Spreadsheets often contain missing data.* 56.1% of the spreadsheets for *C1* and 20.2% of spreadsheets for *C2* contain N/As.

*Spreadsheets sometimes contain color coding.* Conditional formatting is used for 12% of *C1* and 35% of *C2* spreadsheets. It indicates that users are willing to make sense of their data by using visual cues.

Overall, 86% of the spreadsheets we analyzed are compatible with Bertin's method requirements. The typical spreadsheet contains a unique table with 10–37 entries and 6–14 dimensions, far below the limits of BERTIFIER; both entries and dimensions have 1–2 headers and mostly contain quantitative values.

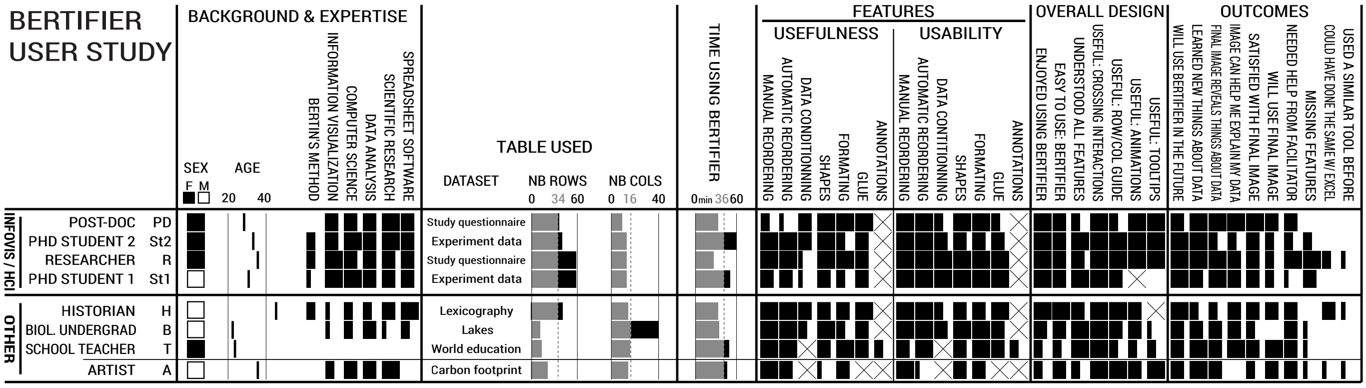



Figure 6.14: Visual summary of participant demographics, datasets, time on task and questionnaire responses. Crosses indicate unused features.

## 6.5 QUALITATIVE USER STUDY

Figure 6.14 presents the results of the qualitative study in a bertified table. The rest of this Section refers to this Figure.

### 6.5.1 *Procedure*

We recruited participants willing to visualize a data table they were interested in. They initially sent us their table of interest. We limited table sizes to $40 \times 15$ (which represents more than 75% of personal tables according to our spreadsheets analysis) to avoid too long sessions. We selected eight participants based on their background, and favored diversity of datasets and tasks. Overall, the average number of columns was 16 (min 9, max 40) and the average number of rows 34 (min 11, max 57). We cleaned up some of the tables with the participant to ensure compatibility with Bertifier.

The session consisted of five phases: i) the participant read short instructions about the session; ii) the facilitator asked the participant to explain why the dataset has been chosen, what was already known about it, and what the hypotheses were; iii) the participant watched a six minutes video tutorial explaining how to use Bertifier and covering all features (the same scenario as in the walk-through Section 6.2); iv) the participant switched to Bertifier in the web browser. We encouraged participants to think aloud and they could ask questions to the facilitator, although they were not guided when using the tool. We took notes on everything the participant said, as well as questions, difficulties, behavior and insights. When participants decided that the work was done using the tool, they could open the final result in a vector graphics editor to finalize it. Finally, participants filled a questionnaire. The session ended with a short (10–15 minutes) semi-structured interview. The average session time was 75 minutes (min 60, max 130), including 29–52 minutes using Bertifier. Summary data is reported in Figure 6.14.

### 6.5.2 *Participants*

Participants could be roughly categorized into two groups of equal size: *Infovis/HCI* and *Other*. In the *Infovis/HCI* category, participants tended to be experts in Infovis, HCI, computer science, data analysis, scientific research and spreadsheet software.

The Post-Doc (***PD***, HCI) wanted to analyze participants' answers to two online user study questionnaires. She did not analyze the data and expected to see differences between the two techniques she wanted to compare, in order to strengthen her quantitative results.

The researcher (***R***, Infovis) also wanted to analyze participants answers to a user study questionnaire, for a project on glyph design [Fuchs et al., 2014]. She had hypotheses, but did not look at the results. She expected to discover interesting patterns that are worth statistical analysis.

The two PhD students (***St1***, Infovis, ***St2***, Infovis/Design) worked independently on the same dataset. Their table contained both qualitative (questionnaire answers) and quantitative (e. g., time, counts) results from an exploratory user study on constructive visualization [Huron et al., 2014]. With 57 rows and 13 columns, their table was the largest of our evaluation. They wanted to find out if their measures were correlated with participant's demographics. They also wanted to group participants according to the way they performed the tasks. Overall they were seeking an informative overview of heterogeneous data that is difficult to understand, and wanted to report the trends in their article.

In the *Other* category, participants had little or no expertise in Infovis, and varying levels of expertise in other domains.

The Historian (***H***) wanted to visualize lexicographical data from a project he already published about the vocabulary used in judiciary texts. He knew the data very well and already analyzed it using statistical methods. His task was not exploratory but confirmatory: he wanted to check that the visualization tool will highlight the same results.

The Biologist undergrad (***B***) was interested in rivers and lakes; he provided a dataset with 40 lakes and 11 characteristics. He knows the topic but he wanted to *"have an overview of the location where there are no large lakes and understand why."* His hypotheses were that there were only a few large lakes in the southern hemisphere, and in highly elevated countries. He expects to challenge his knowledge and discover interesting facts or correlations.

The School teacher (***T***) provided a table with country areas as columns, and education indicators as rows. She presented this table in one of her classes and realized some pupils did not know how to read a table and had difficulties making sense of such heterogeneous numbers.

The Artist (***A***) provided a table with countries and environmental indicators. He had been active in an environmental association and was interested in communicating facts and convincing people. He also wanted to *"have a more global view of [his] own carbon footprint"* by making sense of those numbers.

### 6.5.3 *Results*

Observations, questionnaire responses (Figure 6.14) and interviews indicate that participants with and without Infovis background found Bertifier useful. All were enthusiastic during the session and rated Bertifier as easy to use and enjoyable. They found it useful both for exploration and communication, and for personal and professional use.

We start by analyzing the way participants interacted with Bertifier to identify which feature they used, why they used them, and how it helped them in their exploration, following the order of the bertified table.

*Manual reordering* was the first step for two participants (*A*, *T*) who reordered rows and columns based on their knowledge and the data semantics. Sometimes, it also helped to fix the result of *Automatic reordering* if they judged it unconvincing (*St2*). Some participants used this feature to move away rows or columns that were not interesting (*St1*, *St2*), and *St2* took advantage of the animated transitions by dragging slowly a row over other ones to identify correlations visually. All participants gave the maximum score for *Manual reordering* usability and *Other* participants found it more useful than *Infovis/HCI* ones (R5).

*Automatic reordering* was one the most used features and generated the most insights (R5). Just after reordering, participants discovered surprising facts (e. g., *"I was not expecting to see Portugal in first position according to forest area"*, *A*), generated new questions (e. g., *"I think that if I had the age of the lakes as a dimension, it would be very interesting to see how it correlates"*, *B*), and provided meaningful arrangements of the data (e. g., *"I understand something that was hidden in my previous factorial analysis"*, *H*). All participants but *A* reordered their table locally several times by dragging over a subset of adjacent crossets. They did it when they were satisfied with some groups, but not with the rest of the table. It yielded new findings (*R*, *T*).

*Data conditioning* was mostly used for two reasons: first, to reorder the table according to a subset of rows only. Participants performed this task by reducing to its minimum the *Strength* of the rows they wanted to ignore, using crossing gestures (*PD*, *R*, *St1*, *B*). Second, *St1* and *St2* de-emphasized rows that they wanted to ignore or that they did not want to show in their final image by setting their strength value to 0. Several participants used the *Range* slider to change the scale of a row when an outlier compressed the other values (*St1*, *St2*, *H*, *B*). *Data conditioning* is one of the only feature lowly ranked by several participants. It suggests that R2 is reserved to advanced users. For example, *PD* and *St2* commented that it was very useful to negate several questions of her questionnaire because they were not all congruent and *B* negated a dimension to check if it was inversely correlated to another one; on the other hand, *T* did not like the negative function at all since to use it, she needed the header of the row to be negated too.

*Shapes*: All participants except *A* tried several encodings to get different views on their data and rated this feature as very useful (R3). They encoded rows differently according to their type: *PD* used grayscale to encode frequencies and barcharts elsewhere; *H* chose the black & white barchart because he was interested in perceiving average values; *B* encoded latitude and longitude using lines to compare their position in space, and circles to convey congruent measure with shape such as area and volume. Two participants (*T*, *A*) commented that squinting at the table/blurring their eyes revealed areas and were surprised to realize that they were reordering the table without paying attention to the semantic of the data, but only the shapes.

*Formatting*: All participants resized at least once the rows and columns to get an uniform layout using a dragging gesture. *St2* also used this feature to highlight interesting rows by increasing their height. All participants used separators during their session, both for exploration and for communication purposes. More expected use of separators included adding white spaces for readability (7/8 participants) and separating groups for communication purposes (6/8 participants).

*Glue* was used at least once by all participants except *A* (R6). The first gluing often happened after the participant asked how to move several rows or columns at once in order to save time (*St1*, *H*, *T*). It helped move apart final groups and focus on the rest of the table. *PD*, *B* and *H* found that the interaction with the *Glue* crosset was not obvious but that the animated feedback on the matrix was helpful. Overall, participants found this feature very useful but not straightforward.

*Annotations* were used by *T* only (R7). Other participants either forgot about this feature (*St2*, *H*) or preferred to annotate their image in a vector graphic editor (*PD*, *R*, *St1*, *B*) thanks to the SVG export function (*R:comm*).

### 6.5.4 *Overall Benefits of Bertifier*

All *Infovis/HCI* participants seemed convinced by the overall design of BERTIFIER, and reported finding it very easy to use and enjoyable. Only *H* (and to a lower degree *R*) thought he could have accomplished the same using a standard spreadsheet. Participants in the *Other* group needed more help from the facilitator compared to *Infovis/HCI*. However, no one asked for extra features. The two strongest takeaways from the experiment are about crossets and automatic reordering:

**Crossets** obtained the maximum score from all participants, who almost never clicked individual crossets but applied crossing gestures to series of crossets. They commented it was extremely useful (*R*, *PD*, *St1*, *A*) and powerful (*H*). No participant had difficulties understanding and reproducing the gesture they saw in the video tutorial. They were even sometimes positively surprised to see that the gesture worked on slider widgets (*R*) and for sliders that were previously set to different values (*A*). Surprisingly, all participants found the *Formatting* crossets extremely easy to use but not the most useful. We expected the converse to be true, as formatting seems important to us, but our tools have a low degree of *compatibility*, as the gesture direction to change the size of rows, columns and separators is orthogonal to its effect.

**Human-assisted reordering** coupled with grouping and conditioning was considered an easy and efficient way of exploring the data. *R* said that BERTIFIER was the only tool she knows with reordering capabilities : *"Usually I do the reordering by hand, and do not see a matrix visualization of it."*. *St2* said that automatic reordering allows her to do *"something [she] always wanted to do with [her] table"*. She already used spreadsheet tools for this purpose but found them tedious.

All participants learnt new things about their data. *H* gave the only score below 4/5 for learning new things, as he already analyzed his data thoroughly. For example, *PD* found that the big value of her production is that she *"can clearly see the pairs, that [she] could not see with excel"*; *R* now knows which statistical tests she should run and she identified two groups of participants. They all anticipate several situations where they would use BERTIFIER (min score is 4/5). They consider using it for exploring personal (*St1*, *B*) and heterogeneous (*PD*, *St2*) data, and for communicating findings (*St1*, *A*) and illustrating articles (*R*, *PD*, *B*, *St1*) with the ability to present all their data (*R*, *St2*). *St1* and *St2* later used BERTIFIER to craft an overview of their data and included it in a paper submission [Huron et al., 2014]. *H*, *T* and *A* also see a pedagogical value in the tool. *H* wants to integrate BERTIFIER into the Master's class he teaches as soon as it will be available on-line, both for pedagogical purposes and to provide exploratory analysis of tabular data. *T* believes that the image she created could be useful for teaching lessons when presenting a numerical table of small size is already challenging.

## 6.6 DISCUSSION AND FUTURE WORK

BERTIFIER is the first attempt to remaining faithful to Bertin's original matrix method, while leveraging the power of computers, controlled with recent methods from HCI. This design choice gives us a fresh perspective on Bertin's approach. Our user study suggests that Bertin's method is still valid and useful today, but deserves to be further refined.

*Non-Sequential Analysis.* Bertin was constrained by the technology available at his time, but modern computers now allow to relax some of these constraints. In particular, the method does not necessarily need to follow a linear order: conditioning (R2), choosing the encoding (R3), presenting the table, grouping and reordering (R5) can be done in any order and as many times as necessary to explore the data, generate new hypotheses, and converge towards the best visual image. BERTIFIER can still be improved in this direction, as formatting (i.e., glued groups, separators, and annotations) is not well-integrated with reordering.

*Semantic Reordering.* Our study stresses that Bertin's famous "Town dataset" is only an idealized example meant to illustrate the benefits of data-driven reordering. Many datasets do not yield clean clusters, and users sometimes prefer to preserve the natural hierarchy and/or ordering of dimensions in order to facilitate reading, independently from the data patterns. Many datasets are also heterogeneous, leading users to use a mix of data-driven reordering (automatic and manual) and semantic reordering (manual only). Overall, BERTIFIER's essential value is to support the presentation of tabular data in a compact and legible fashion, and automatic reordering is only one tool among others.

*Visual Encodings.* Not all of Bertin's original visual encodings may be optimal and easy to interpret. Some participants expressed doubts about the effectiveness of certain encodings, although personal preferences varied a lot. Some participants also felt the provided encodings did not meet their needs for specific data types, such as dimensions with values above and below a baseline (e. g., temperatures), or dimensions with only a few possible numerical values that need to be easy to discriminate. Because BERTIFIER is open source, new encodings can easily be added, tested and empirically compared with Bertin's original encodings. A strong need was also expressed for legends, that currently need to be added in an external authoring application.

*Qualitative Data.* Supporting qualitative or categorical data is another key limitation of BERTIFIER. Bertin proposed categorical encodings for maps [Bertin, 1975] but not for matrices. He typically treated qualitative dimensions with $n$ possible values as $n$ binary dimensions [Bertin, 1975], but this approach does not scale to large $n$. One problem with categorical encoding is that it requires users to explicitly map each possible data value to its visual representation (e.g., a specific color or texture). Previous work showed that this operation can in some cases be automated [Lin et al., 2013], but users still need a way to tweak the results. Implementing such a feature poses non-trivial problems of HCI design, and may possibly require the addition of other tools than crossets.

*Scalability.* Like Bertin's physical matrices [Bertin, 1975], BERTIFIER does not support very large tables. Although our previous analysis suggests that personal spreadsheets are typically small, large tables exist and can cause problems to BERTIFIER. First, they may require scrolling. Although scrolling is supported, crossets cannot be crossed beyond the viewport. One solution is to zoom out (also supported), but interaction becomes difficult as widgets become smaller. Another approach could be to let users collapse or aggregate groups of rows or columns, or (as suggested by *R*) split a table into several sub-tables to be manipulated independently. Finally, one could consider using arbitrarily large high-resolution displays. Although in principle very large tables can be shown even on a 30″ screen, another bottleneck lies in performance, as user interface responsiveness quickly decreases with SVG scenegraph complexity.

Participants from the user study suggested various other improvements, such as being able to rotate text headers, set cells' background color, and split the table into subparts that can be reordered separately. Many other extra features can be considered that would support more effective communication, such as adding legends to individual dimensions, labels to individual cells, and various other graphical decorations. Finally, we are considering adding support for interaction history and revisitation based on small multiples as proposed by Bret Victor [Victor, 2013a], as well as the ability to export specific states as URLs.

## 6.7 CONCLUSION

We presented BERTIFIER, a tabular visualization authoring tool based on Jacques Bertin's matrix analysis method. BERTIFIER matches and extends the requirements stated and implied by Bertin. We contributed a new interaction technique—the *crosset*—that lets users quickly and easily apply commands to series of rows or columns. Crossets could be used in other Infovis systems (e. g., for dynamic queries), but further evaluations are needed to better assess their strengths and weaknesses. We also introduced *visual reordering*, an approach that lets users apply and tune automatic reordering algorithms in a WYSIWYG manner. This approach fills a gap between flexible but often tedious manipulations, and fast but often unsatisfactory automatic reordering methods.

In his review of matrix reordering methods, Liiv concludes that *"seriation cannot be considered ubiquitously usable, until implemented and shipped as a standard tool in any spreadsheet and internet browser for enabling such analysis. Then one can say that seriation and matrix reordering is usable. That is the main future goal for seriation."* [Liiv, 2010]. BERTIFIER is a significant step towards this goal, as our user study suggests it makes it possible for both scientists and a wider audience to explore, analyze and interpret their data, as well as to communicate their findings by visual means. We will continue to extend BERTIFIER based on user feedback, and further hope that its design will inspire extensions to popular commercial tools such as spreadsheet software, so that Bertin's method finally becomes accessible to a wide audience and empowers people with data analysis and communication tools that were so far only accessible to a small number of specialists.

### 6.7.1 *Design Guidelines*

Designing BERTIFIER involved many design iterations. Here we discuss the value of internal consistency, the importance of identifying the objects of interest, the value of many widgets, and slow interaction and human-steered algorithms. We refer to the summary of principles, benefits and challenges summarized in Table 2.1.

The first observation is the analogy between BERTIFIER's original method and the Infovis pipeline (1.5). Indeed, the several steps of the methods are similar to the ones of the Infovis pipeline and Bertin acknowledged that hand-made and physical visualization required interaction, leading to what he called "mobile images". One of the strength of our system is the non-sequential approach of the method, thus the ability to alter the Infovis pipeline at all stages by editing the data in a spreadsheet and then interacting in a unique view allowing to go back and forth between the different stages in a fluid and continuous manner. Actually, the only stage that is missing in BERTIFIER is data editing, that could be solved by double clicking in a cell for example and is part of future work.

For future work, we identified various improvements to extend the capabilities of BERTIFIER but we need to keep in mind the tradeoff between power and simplicity. For example, inline editing of the data is not straightforward to implement as it would require to change the interaction mode, altering the fluidity of the interface (P18).

Indeed, such interaction would be helpful and complete the system but is not compatible with the interface design whose core advantage is its high internal consistency. All actions on the table are performed through crossets, a unique and generic interaction metaphor improving learning, memorability, and retaining mastery (B3, B12). Moreover, because the crossets are augmented standard widgets that embed their default well-known interaction modalities, it has a high external consistency helping transfer of training (B15).

▷ Designing a fully internally consistent interface requires careful decisions and may require discarding functionalities.

OBJECT OF INTEREST DRIVEN DESIGN For BERTIFIER, internal consistency raises issues because the interface is designed around particular objects of interest. Indeed, the objects of interest in traditional tabular visualizations are the cells of the table. Following Bertin's method, the objects of interest in BERTIFIER are the rows and the columns. All actions are performed on rows and columns, making possible to layout the crossets around the table. This layout allowed us to implement the drag & drop of rows or columns by dragging cells. Again, to remain fully consistent with the interface design, dragging a cell is locked in one direction according to the mouse movement in order to drag either the corresponding row of the cell, either its corresponding column. Internal consistency makes the interface efficient (B13) to reduce the efforts required by the user (B10).

Because the objects of interest are rows and columns, crossets introduce spatial indirectness (P11). We limited spatial indirectness by duplicating crossets such that the spatial offset is null according to one of the two dimensions of the space. However, introducing spatial indirectness is not always a bad thing, as noted by Beaudouin-Lafon and we think that crossets for tables are a good example of beneficial spatial indirectness . The congruence of the interactions also suffers from the low degree of compatibility of some actions. For example, to modifying the width of a column, the user has to drag the slider thumb up and down while a horizontal dragging would be more congruent. On the other hand, crossets have a high degree of integration (P13).

▷ The whole interface design depends on the object of interest identification.

▷ Internal consistency may conflict with congruence of the interaction.

MANY WIDGETS CAN BE BENEFICIAL Usually, having too many widgets is not recommended (C15). However, the multiplication of crossets is a strength of Bertifier. Indeed, it allows for tackling the difficult challenges of applying an action on several objects of interest by a unique gesture-based interaction (C7, C8). However, this strategy involves targeting issues (C6): it is not possible to drag over a crosset when it is outside the viewport. Moreover, as the number of widgets is proportional to the number of objects of interest in the visualization—rows and columns in Bertifier, the rendering performances of the computer limit the scalability of the visualization.

▷ Many widgets can be beneficial when carefully designed, but scalability may be limited.

SLOW INTERACTION AND HUMAN-STEERED REORDERING As detailed all along this chapter, few works have been dedicated to the tradeoff between fully automatic and fully manual reordering methods, although recognized as important in the community (C17). The interactions we provide are a step towards this direction. The user study showed that users without any statistical background successfully used human-steered reordering methods, certainly because we used the metaphor of visual reordering that does not require any mathematical background from the users.

Interactively steering the reordering algorithm generated many insights that we believe—in agreement with Bertin—an automatic method would not have revealed. By slowly interacting with the data, users discovered unexpected facts and were fully engaged in the exploration process, reporting playfulness and enjoyment.

▷ Understanding a dataset takes time, and spending time manipulating a dataset generates insights.

▷ Visual human-steered algorithms and slow interaction generate insights, engage users, and does not require a particular background.

### 6.7.2 *Crossets: Discussion*

Crossets are the basic interactive component in Bertifier. Their main advantage is the enhancement of standard widgets by exploiting crossing gestures while avoiding the main drawback of the interaction technique: steering. Indeed, locking the crossing trajectory in the orthogonal dimension of the crosset axis avoids steering the gesture, a slow motor task [Accot and Zhai, 2002]. Letting the user deviate her trajectory while avoiding the selection of undesirable crossets by mistake makes the interactions faster and errorless.

Crossets are particularly well suited to the table layout and have been fount efficient and easy to use by all participants. In this subsection, we discuss the limitations of crossets and to which extent the technique applies beyond tables.

#### 6.7.2.1 *Limitations*

The main limitations of crossets are the real screen estate they need, the requirements that similar crossets are adjacent, and the interface stability when crossets impact its layout.

REAL SCREEN ESTATE Crossets must be visible on screen to be reachable. In BERTIFIER, this may be a problem when the table is too large, requiring to perform several gestures coupled with page scrolling. However, zooming out solves the problem in most cases. A promising perspective we consider as future work consists of making crossets transients so that they only appear when the gesture is activated and disappear once the gesture is over.

NON-ADJACENT CROSSETS The design of crossets requires the widgets to be laid out in grid, and crossets are restrained to these (numerous) interfaces: similar crossets must be adjacent and adjacent crossets must have the same values domain. Assigning the same value to non-adjacent crossets is a clear problem. The crossing gesture being orthogonal to the crossets axes, there is no way of avoiding a crosset on the path. Adding a key modifier to "skip" a crosset may solve this issue, but it would add complexity to the interaction while one of the strengths of the technique is its simplicity, and it would require the user to switch modes (P18). We get around the problem by using discrete crossets: the low number of possible values allows for easily attributing the same value to distant crossets, visually. In BERTIFIER, another way of solving this problem consists of grouping rows and columns by drag and drop to make them adjacent. More generally, movable crossets allow for making them adjacent to each others before crossing.

INTERFACE STABILITY Finally, we illustrated the problem of interface stability in BERTIFIER with several examples. To ensure interface stability, we adopted the strategy of providing immediate but incomplete visual feedback as a preview of the result and apply layout modifications once the interaction is over. This way, crossets remain compatible with the interface stability, but at the cost of introducing temporal indirectness (P12).

### 6.7.3 *Generalization*

Crossets extend previous work on crossing-based interfaces in several ways [Accot and Zhai, 2002; Apitz and Guimbretière, 2004; Baudisch, 1998]. One problem with such interfaces is that they require steering, a slow motor task [Accot and Zhai, 2002]. Crossets do not have this problem since the command is selected on mouse press, after which the user is allowed to freely deviate from a straight trajectory. As far as we know, crossing gestures have never been applied to manipulate multiple sliders at once, and have never been used to interact with tables. Crossable widgets open promising perspectives, and we hope more interfaces will take advantage of this technique in the future.

We applied crossets to a table formatting tool, but many applications would benefit from crossing widgets. Dynamic queries [Ahlberg et al., 1992] often involve changing the value of several widgets simultaneously. For example, when looking for an apartment, it may be interesting to simultaneously change the value for "distance to A" and the value for "distance to

B" to specify that both distances are equally important. Crossets may also be efficient to specify a level of gray in the RGB color space by manipulating the three sliders in one gesture. However, the current implementation is not compatible with the frequent case where the dimensions associated to adjacent sliders are different. Nonetheless, dimensions are often normalized, allowing for similar request on semantically different sliders (e. g., heterogeneous better life index indicators are normalized in $[0, 1]$). Conversely, it may happen that assigning identical values to similar dimensions does not make sense: it depends on the semantics of the data.

Finally, a promising perspective would be to allow unconstrained 2D gestures to assign in a unique gesture different values to different crossets by drawing a trajectory in the plane. However, as for avoiding adjacent widgets on the trajectory, it would require activating a new mode and add complexity to the interaction.

## 6.8 ADDITIONAL MATERIAL

Additional material is available at `www.aviz.fr/bertifier`. It contains the BERTIFIER prototype; reproductions of historical examples crafted using BERTIFIER; links to software listed in Section 6.1.7 and a detailed explanation of the calculation of the scores illustrated Figure 6.3; data from the user studies illustrated Figure 6.14; additional related work for the interested reader.

## 6.9 ACKNOWLEDGEMENTS

We thank authors of previous systems and their past collaborators for providing us with material for our research: Günther Sawitzki, Harri Siirtola, Patrice Mitrano, Philippe Lecarpentier, Joël Le Fourn, Gloria Villarroya and Marie-Françoise Durand; Thanks to Emanuelle Giry and Denise Ogilvie for organizing our visit of Bertin's archives in the Archives Nationales; Yvonne Jansen for providing us with numerous datasets; Yvonne Jansen and Benjamin Bach for their early feedback on the paper; Jeremy Boy for his feedback on the visual design of Bertifier; Jeremy Boy and Nadia Boukhelifa for their voiceovers; Fanny Chevalier for convincing me to start the project.

# 7 GUIDELINES AND PERSPECTIVES

I presented in this thesis several interaction techniques for various Infovis systems. Each of the previous chapters illustrates how Infovis can leverage direct manipulation principles and recent advances in HCI to make visualizations more efficient, easier to use, and more engaging.

In this chapter, I first discuss the challenges issued from direct manipulation that this thesis addresses. Then, I discuss empirical observations drawn from these contributions highlighting the benefits of designing interaction techniques opportunistically. Based on this experience, I emphasize several guidelines for designing post-WIMP interactions for Infovis that are not well recognized in the field, raising new challenges for the community. I discuss the importance of object of interest-driven interaction design, the importance of playfulness and aesthetics to enhance user's engagement and experience, and the importance of slowing down the exploration process to generate insights.

As perspectives for future work, I detail the two grand challenges that are the discoverability of interaction techniques and the need for a standard for interaction mapping.

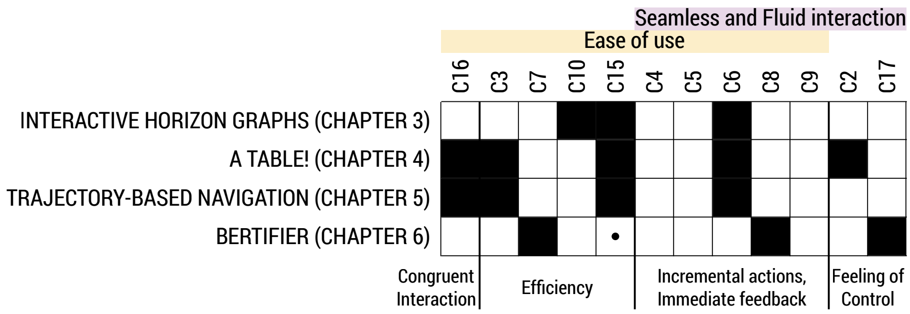



Figure 7.1: Addressed direct manipulation challenges by chapter. Black squares indicate addressed challenges and the black dot indicates that a pitfall was used in contradiction, as a benefit.

## 7.1 ADDRESSED CHALLENGES

I designed interactions based on the principles listed in Section 2.1, to get the benefits of applying these principles. I paid attention at designing new interaction techniques dealing with several of the direct manipulation challenges, in the context of Infovis. I do not discuss here C1, C11, C12, C13, C14, and C11 as they are related to learning, beyond the scope of this thesis. However, I will discuss later these challenges as they are related to discoverability. Although most of the challenges have been explored in at least one chapter of this thesis, I did not tackle the challenges of performing concurrent actions (C9), referring to previous actions (C4), and scheduling actions (C5).

Figure 7.1 illustrates which challenge was addressed in previous chapters. I mainly focused on designing *efficient* (C3, C7, C10, C15) and *congruent* (C16) interactions while improving the *feeling of control* (C2, C17), and tackled two challenges related to *incremental actions and immediate feedback* (C6, C8).

In Chapter 4 and Chapter 5, I faced the challenges of designing both congruent and efficient interaction techniques. I presented one-dimensional and two-dimensional interaction techniques to navigate in three-dimensional information spaces (C16), the third dimension being time. These two chapters also illustrate cases where the difficulty lies in the task domain (C3) and emphasize the difficulty of allowing to perform a broad range of tasks through a unique interaction modality. Chapter 4 also provides evidence that expert performance can be improved by direct manipulation (C2), and Chapter 3 addresses the challenge of specifying precise values (C10). The interaction techniques presented in Chapter 3, Chapter 4, and Chapter 5 are also examples of post-WIMP interactions where the user directly manipulates the objects of interest, no widget at all being used (C15). Conversely, Chapter 6 contradicts this challenge as the BERTIFIER interface features a large number of widgets.

I Addressed the challenge of targeting unseen objects (C6) from two perspectives. In Chapter 3, the values zooming metaphor allows for making visible time series attributes that are difficult to perceive for LSV datasets, meantime tackling the challenge of specifying precise values (C10). Similarly, in Chapter 4, the VIZ-RANK technique shows additional information by representing ranking charts of the objects of interest, making trends visible and comparison possible. In Chapter 5, I addressed this challenge from another perspective by investigating the case of targeting objects of interest that are not visible on the viewport. The provided solution consists of maintaining still the manipulated object while the visual representation around moves, making it possible to navigate in tables larger than the viewport.

Finally, Chapter 6 brings a solution to targeting adjacent groups of objects (C7) and applying actions on several objects of interest by a unique interaction (C8) thanks to crossets, widgets augmented with crossing capabilities. Chapter 6 is also a step towards making algorithms human-steerable (C17) in order to make these algorithms accessible to a large audience.

## 7.2 DESIGN GUIDELINES

I proposed design guidelines at the end of each chapter. Some of the remarkable lessons are that efficient interaction techniques can unify existing visualization techniques (Chapter 3), that interaction techniques themselves can be unified in more versatile ones (Chapter 5), and that opportunistically adding interaction to visual representations may lead to unexpected discoveries (Chapter 3). In this section, I discuss important aspects of interaction for Infovis that are not well recognized in the community but that one or several chapters of this thesis highlight.

### 7.2.1 *Object of Interest Driven Interaction*

Instrumental interaction measures four criteria (P11, P12, P13, P14) according to the coupling between an instrument and the associated object of interest. For example, measuring the spatial indirection (P11) of an instrument involves knowing the location of the object of interest. Thus, identifying the object of interest is an essential step when designing interaction, as previously emphasized (e. g., [Jansen and Dragicevic, 2013]) because the directness of an interaction technique depends on the object of interest.

Chapter 5 highlighted the importance of identifying the objects of interest and emphasized the fact that this is not always an obvious task. For example, defining the visual representation of data points as objects of interest in a scatterplot is straightforward. Conversely, in ranking tables, the objects of interest are table cells. The interaction technique then requires a visual transformation of the object of interest (cell to data point) so that it is compatible both with the table layout and with the motion trajectory representation. The interaction technique we proposed in Chapter 5 has a high degree of compatibility (P14) and a high semantic directness both in the gulf of execution (P5) and in the gulf of evaluation (P6); it is based on physical actions (P2) directly on the object of interest (P11); according to the dragging strategy, these physical actions have a medium to high degree of integration (P13). The general implication for designing integration satisfying the congruence requirements resulting in a natural mapping to the task (P8) is then:

G1 Identifying the objects of interest is the first consideration to design congruent interaction.

Chapter 6 highlighted several design implications of objects of interest-driven interaction design. Indeed, the objects of interest in traditional tabular visualizations are the cells of the table while in BERTIFIER the objects of interest are the rows and the columns, following Bertin's method [Bertin, 1975]. For example, to benefit from a fully internally consistent interface (B12, B13, B14), cells of the table cannot be dragged but the corresponding row and column can instead. Another design implication we discussed is the spatial indirectness (P11) introduced by crossets according to one of the two dimensions of the space, thus reducing the congruence of the interaction. However, crossets for tables are a good example of beneficial spatial indirectness because they do not interfere with the interactions performed directly on the table. We also saw that crossets have a high degree of integration (P13), but sometimes a low degree of compatibility (P14).

G2 Reducing the congruence of the interaction by reducing both the degree of spatial directness and the degree of compatibility may be beneficial to increase the internal consistency of the interface.

### 7.2.2 *Congruence vs Versatility*

Dragicevic et al. argue that *"designing for direct manipulation involves matching user's gestures with the observed visual motion"*. Indeed, this argument is in line with the principles for designing direct interactions. However, if the direct manipulation of objects of interest in DimP [Dragicevic et al., 2008] is highly congruent to spatial tasks, the technique is not efficient for other temporal tasks such as comparing trends and quickly reaching a given time. This raises the tradeoff between congruence and versatility, as interaction techniques cannot be congruent to every possible task.

Chapter 4 illustrates this tradeoff: DRAG-CELL is very congruent to a small subset of tasks (inverse tasks) by interacting with the values domain instead of the time domain, but inefficient for the majority of temporal tasks. On the other hand, VIZ-RANK is a more versatile technique as it allows for performing efficiently a broader range of tasks, but at the cost of being less congruent. Each task having a different articulation, it is difficult to imagine a unique interaction that would be congruent to all task categories. We tackled this tradeoff in Chapter 5, by merging DRAG-CELL and VIZ-RANK into a unique interaction technique allowing to perform a wide range of time-related tasks. We also maximized the congruence of the interaction (P19) by reducing both temporal indirectness (P12) and spatial indirectness (P11).

Finally, we saw that the versatility of the interaction technique could be increased at the cost of reducing its cognitive congruence. Indeed, in Chapter 5 we discussed the benefits of limiting the degree of integration (P13)—a factor of congruence—to 1/2 for time navigation to avoid user's steering, a slow motor task. In Chapter 6, we made the choice of increasing both the versatility of the interaction technique and the internal consistency of the interface, at the cost of reducing the congruence of the interactions by reducing their semantic directness (P5) and by introducing spatial indirectness

(P11). Indeed, increasing the congruence of the interaction would have required different interaction modalities, reducing both the efficiency of the visualization (P7, B14) and the feeling of control of the user (B13, P18).

G3 Designing an interaction technique involves balancing its versatility and its cognitive congruence.

### 7.2.3 *Engaging Interaction: playfulness and aesthetics*

Dix and Ellis argue that a visually engaging interface or visualization invites experimentation and use. However, they also state that aesthetics should not be a priority for Infovis systems [Dix and Ellis, 1998]. Conversely, Jennings defines engagement as *"the appropriate level of complexity and mystery that will keep the user engaged, but it is immediate positive perceptual judgement of an environment that will entice the user toward exploration and active discovery."* [Jennings, 2000]. Frohlich calls gracefulness the property of an interaction that would be recognized as *"pleasing or attractive, especially in form, movement or action"* [Frohlich, 1993]. Pike et al. add that the dialog between the user and the interface should be playful in order to encourage exploration [Pike et al., 2009] and stimulate user's enjoyment (B8).

In Chapter 4, we designed novel interactions following Bederson principles to preserve user's flow, to focus on tasks and prevent interruption [Bederson, 2004]. Interactions were playful and enjoyable, as indicates the feedback from the participants to the evaluation. Despite the fact that seamless and playful interaction may slow down the users to perform tasks, we showed that performance increases using these two interaction techniques.

G4 Fluid and engaging interaction does not always contradict efficient interaction.

Chapter 4 also illustrates the difficulty to engage users in performing tasks. Soccer enthusiasts were eager to explore the data and spent time on the page. However, without any incentive, they were not willing to complete tasks and even less interested in providing questionnaire answers and feedback. It is then important to separate out user's engagement to freely explore a visualization—where no incentives are needed—and user's engagement to perform low-level tasks—where incentives seem mandatory. Indeed, in Chapter 6, we saw that participants to the user study were fully engaged in the exploration process, reporting playfulness and enjoyment, while no incentive was provided. A possible reason for being that much engaged in the exploration is that participants realized by themselves that the insights they may generate were incentives: they were analyzing their own data, they had their own tasks, and they were free to explore it with no performance or time constraint. Moreover, the careful interface design allowed them to quickly master the interactions thanks to its high external (B15) and internal consistency (B12, B13, B14), giving them the illusion to manipulate directly their data, not the interface, thus giving them a sense of control (B9).

G5 Playful and pleasant interactions engage users as they quickly master the interface.

Chapter 5 presents an interaction technique respecting the principles for a seamless and fluid interaction, featuring incremental actions and immediate feedback, increasing the feeling of control, and favoring playfulness and engagement. The resulting guideline is that seamless and playful interaction that engage users and make them feel in control requires forsaking standard widgets. However, Chapter 6 contradicts this statement. Indeed, on one hand, users clearly reported that the interface was engaging, easy to use, and playful as a game. On the other hand, the interface is exclusively made of crossets, and the large number of crossets is actually the strength of the interface. In this chapter, we advocated that many widgets can be beneficial when carefully designed, thus that widgets are not (always) evil. This divergence between Chapter 5 and Chapter 6 mitigates challenge C16, as both strategies can lead to engaging interfaces.

G6 Seamless and playful interactions that engage users and make them feel in control require either forsaking standard widgets or designing more efficient widgets.

### 7.2.4 *Slow Interaction*

In Chapter 6, we investigated the important challenge of designing interactions for human-steered algorithms (C17). The user study showed that users without any statistical background successfully used human-steered reordering methods, certainly because we used the metaphor of visual reordering that does not require any mathematical background from the users. Slowing down the exploration process led to both user engagement and unexpected discoveries.

Playfulness and flow are related to what I call *slow interaction*, an extremely important but unexplored aspect of Infovis that few researchers have started to mention. Slowing down interaction goes against the general trend, which is usually concerned by optimizing time and performance and make the users save precious time. In one of his blog articles, Stephen Few raises this issue of trying to achieve everything faster [Few, 2013]. This is particularly true for Infovis, and especially for data exploration whose ultimate goal is to generate insights. Indeed, insights cannot be measured in terms of efficiency, correctness or time; insights are unexpected, complex, built over time, relevant, and qualitative [North, 2006]. This aspect of interaction leads to challenging research as it goes against the general trend and may appear as being a step back at first sight. In his EuroVis 2014 Capstone, John T. Stasko emphasizes this aspect: while in HCI, techniques are evaluated based on *benchmark tasks*, visualization is *"about exploration and understanding"* and characterized by the *"vagueness or absence of specific tasks."* [Stasko, 2014].

G7 Slowing down the exploration process should be considered at its right value as slow interaction engages users and helps generating insights.

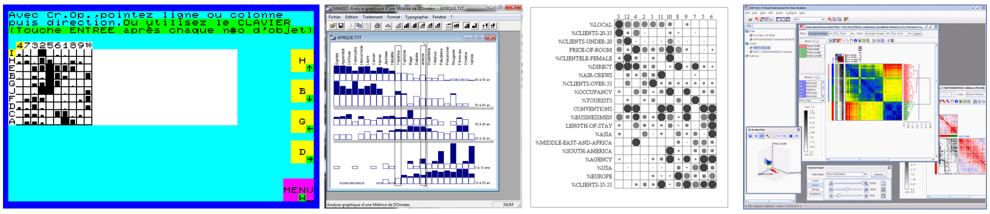


Figure 7.2: Examples of compared software. From left to right: Matrix (TO7/Nanoréseaux), AMADO (Windows), the reorderable matrix (Java web start), GAP (Windows).

| | Indirection | Integration | Compatibility |
|---|---|---|---|
| Menus | -- | +/- | -- |
| Toolbars | - | + | + |
| Dialog boxes | -- | +/- | - |
| Property boxes | + | +/- | - |
| Handles | ++ | ++ | ++ |
| Window titles | ++ | + | + |
| Scrollbars | + | - | - |
| Keyboard shortcuts | + | + | - |
| Drag & drop | ++ | ++ | ++ |

Figure 7.3: Characterizing qualitatively interactive components according to the instrumental interaction model

## 7.3 PERSPECTIVES AND CHALLENGES

Several of the guidelines I provide in the previous Section, such as slow interaction, are also perspectives for future research. In this section, I emphasize three bigger challenges for interaction for Infovis that I did not discuss yet but that some chapters raised. First, I stress the need for methods to compare interactive systems. Then, I discuss the discoverability of interactions. Finally, based on these two first discussions, I lay emphasis on the need for an interaction mapping standard.

### 7.3.1 *Comparing Interactive Systems*

In Chapter 6, we expressed the need for analyzing, quantifying, and comparing interactive systems from two perspectives. Comparing features availability, and comparing *How* features are accessed, in terms of interaction style. Comparing features availability is not a difficult task: a software either proposes the feature or not. However, comparing quantitatively heterogeneous features in software ranging from conversational interfaces to direct manipulation interfaces (e. g., Figure 7.2) in terms of interaction styles is difficult.

Many models exist for the comparison of user interfaces, such as the UAN model, centered on users and tasks [Hartson et al., 1990], and the GOMS model that decomposes user interactions into elementary actions [Card et al., 1983]. Much less models are dedicated to the comparison of interface interaction styles.

Table 7.1: Numerical Scoring scheme for evaluating interaction styles.

| Interaction style | | Spatial directness | Temporal directness | Degree of compatibility |
|---|---|---|---|---|
| Script or programming language | | 0 | 0 | 0 |
| Conversational | | 0 | 1 | 0 |
| Sequences of menus, text fields, dialog boxes | Only text | 0 | 1 | 0 |
| | Icons and Widgets | 0 | 1 | 2 |
| Contextual menus | Only text | 4 | 3 | 0 |
| | Icons and Widgets | 4 | 3 | 2 |
| Hot keys | | 0 | 5 | 0 |
| Widgets far from objects of interest | | 0 | 5 | 2 |
| Widgets close to objects of interest | | 4 | 5 | 2 |
| Direct manipulation of objects of interest | | 5 | 5 | 5 |

Beaudouin-Lafon showed that most interactive components of an interface can be characterized *qualitatively* according to the instrumental interaction model [Beaudouin-Lafon, 2000] (Figure 7.3). However, visualizing a qualitative description of components is difficult, and this method only analyzes individual components, not a whole interface. To compare complete interfaces, Beaudouin-Lafon counts the number of interactive elements (e. g., buttons, menus) it contains [Beaudouin-Lafon, 1997]. This method is useful and quantitative, but tedious as it involves referencing each widget of the interface. Moreover, this strategy is well suited for comparing, for example, two successive versions of a software and two similar software (e. g., Microsoft Word and Open Office); but it does not allow for comparing heterogeneous systems in terms of interaction style because it only captures WIMP and post-WIMP interfaces. To better quantify heterogeneous systems, we identify eight categories of interaction styles:

1. script or programming language. May require a compilation.
2. Conversational: command language, dialog controlled by the system.
3. Sequences of menus, text fields, and dialog boxes where the system guides the user, and the dialog is controlled by the system
4. Sequential contextual menus, where the dialog is controlled by the user.
5. Hot keys, not requiring pointing.
6. Widgets far from objects of interest, with immediate feedback (e. g., menu bars).
7. Widgets close to objects of interest and immediate feedback.
8. Direct manipulation: physical direct actions on the (representations of) objects of interest.

We then used a scoring scheme based on instrumental interaction to characterize these interaction styles, presented in Table 7.1. Many more direct manipulation principles could be quantified this way, but the three dimensions we used already provide a good overview of the directness of an interface. The scores associated to each interaction style are also subjective and only indicative. Also, we did not include the degree of integration in the scoring scheme as this dimension depends on the input device, not the interface itself. Despite being rough, this scoring scheme captures the directness of interfaces ranging from script language to post-WIMP direct manipulation. Because most interfaces feature several interaction modes, we compute each indicator of the scoring scheme as follows:

$$D = \frac{1}{n} \sum_{k=1}^{n} F_k I_k, k \in [1, n] \tag{7.1}$$

$D$ being the computed degree; $n$ the total number of evaluated functionalities; $F_k$ each functionality of the interface; and $I_k$ the interaction style to perform the functionality $F_k$. It simply corresponds to the average of the directness scores of all functionalities of the interface. When several interaction modes trigger the same functionality, we keep for each degree the highest score. We also added to these three indicators an internal consistency indicator, in terms of interaction styles:

$$Consistency = 0.5 \times (N/(\#SI \times \#W)) \tag{7.2}$$

$N \leqslant n$ being the number of functionalities, $\#SI \in [1, 9]$ the number of interaction styles used in the interface, and $\#W$ the number of interactive components of different interaction types that trigger the evaluated functionalities. Finally, we added a subjective indicator of *usability* taking into account the discoverability of the interface, its ease of use, and more generally the user experience.

Based on this scoring scheme, we were able to analyze, compare, quantify, and finally visualize heterogeneous systems ranging from conversational to WIMP interfaces in Chapter 6 (Figure 6.3), allowing us to classify interfaces into categories, according to their interaction style.

This scoring scheme is an attempt to *quantify* interfaces in terms of interaction styles. More indicators could be added to make the comparison more complete, but one of the advantages of this scoring scheme is its simplicity, allowing for quickly comparing interactive systems.

### 7.3.2 *Interaction Discoverability*

The second perspective for future research is interaction discoverability. In Infovis, discoverability can mean the ability to discover new facts, and generate insights. Here I discuss discoverability from an HCI perspective, which has been defined in several ways.

#### 7.3.2.1 *What does Discoverability Mean?*

For Norman and Nielsen, discoverability is deterministic and ensures that *"all operations can be discovered by systematic exploration of menus"* [Norman and Nielsen, 2010]. Antle et al. define discoverability as how likely it is that a particular group of participants will discover an input mapping by chance [Antle et al., 2009]. In both cases, the authors consider discoverability as the first phase of learning. On the other hand, Agarawala and Balakrishnan use the term discoverability to characterize user's discoveries of complex interactions on their own, *after* having been taught a small set of initial basic interaction techniques [Agarawala and Balakrishnan, 2006]. Here, discoverability is not considered at the initial trigger to learning, but as a factor of mastering an interface. Similarly, Hosio et al. argue that discoverability is a factor of engagement and long-term use by the users that can have a drastic effect on perceived application efficiency [Hosio et al., 2013]. Finally, the term is sometimes clearly misused. For example, Kane et al. argue that interactions are discoverable on touch screens because the users directly manipulate items on the screen [Kane et al., 2008], mixing the directness of the interaction and its discoverability. The enumerated principles, benefits and challenges of direct manipulation in Table 2.1 related to learning also emphasize the different stages of learning. Using metaphors (P4), improving transfer of training (B15) and taking into account user's prior knowledge (C1) are related to the starting point of learning. Providing a natural input mapping (P8) and designing congruent interactions (P19) are related both to the starting point of learning and to recall. Consistency with the world beyond computing (B16) and internal consistency (B12) are related to recall. Finally, confidence in retaining mastery (B3) is related to expertise and long-term use.

The first important point about discoverability is thus the need for properly defining the term. Here, I consider discoverability as the starting point of learning, following Norman and Nielsen and Antle et al. definitions, and assuming that interactions can be discovered either by exploring the interface, either by chance.

Chapter 6 illustrates the discoverability of interactions by exploring the interface. Because Bertifier is highly consistent, both externally and internally, users can transfer their knowledge (B15) and quickly learn all the functionalities of the interface (B12). In this case, discoverability is related to previous knowledge of the user that reduces the amount of learning required to learn the meaning of the components of the graphic representation [Shneiderman,

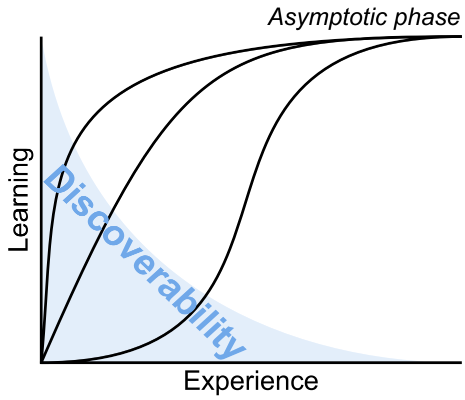



Figure 7.4: Discoverability on learning curves.

1983]; and *Learnability* and *memorability* depend upon the user's prior knowledge [Hutchins et al., 1985].

Chapter 4 illustrates the discoverability of interactions by chance. In the crowdsourced study, users discovered the new interaction techniques by themselves, when they were eager to explore the data, but several participants commented that they discovered the interaction by chance the first time.

#### 7.3.2.2 *Discoverability on the Learning Curve*

In this thesis I assumed that the user was on the asymptotic phase of the learning curve, thus knew the available interactions. Indeed, it is a common practice for evaluating new techniques in HCI as participants are usually trained. This was the case in Chapter 3, where participants to the user study practiced the interactions before performing actual trials.

The learning curve [Hall and Titchener, 1903] describes how knowledge is acquired over time as a *"rapid rise at the beginning followed by a period of retardation, and [...] thus convex to the vertical axis."* [Hall and Titchener, 1903]. A learning curve can have a variety of shapes (e. g., exponential, sigmoid, steep and shallow [Leibowitz et al., 2010; Newell and Rosenbloom, 1993]) but is usually an exponential rise where the beginner has a steep curve and the expert reaches the asymptotic knowledge. However, no matter the shape of the curve, if the user has no initial knowledge of a system, her learning curve has to be initiated.

Figure 7.4 shows the role of discoverability on different learning curves. Considering discoverability as the starting point on the learning curve to discover the communication possibilities with an interface, the more the interactions are discoverable, the steeper the learning curve can be, and the faster the user may reach the asymptotic phase of learning.

#### 7.3.2.3 *Factors of Discoverability*

To trigger interaction discoveries, user interfaces make use of affordances according to Norman's definition, where the design aspect of an object suggests how the object should be used [Norman, 1988]. Seto et al. add that affordances and animations increase the discoverability of the interactions in tabletop settings [Seto et al., 2012]. Discoverability is a clear challenge for gestural interfaces, as *"So far, nobody has figured out how to inform the person using the app what the alternatives are"* [Norman and Nielsen, 2010]. However, discoverability is also a challenge for on-desktop visualizations. In this thesis, we empirically observed several factors of discoverability that are not related to affordances.

- *External consistency:* in Chapter 6, we observed that designing an interface based on well-known conventions (widgets) helped transfer of training thus discoverability (sometimes by accident).
- *Internal consistency:* in Chapter 6, we observed that internal consistency and versatility helped used discover all features of the interface: once one interaction is triggered, all the interactions are learned.
- *Engagement:* in Chapter 4, we observed that the more the user was engaged in the dataset and the visualization a-priori, the more she discovered the interactions by accident.
- *Congruence:* in Chapter 5, we guessed that discoverability is made easier if the interaction technique is cognitively congruent to the user's intent.

To conclude, discoverability is a promising challenge for the community. In the context of Infovis, discoverability is a factor that may reduce the cost capturing users' time investment to select and learn the visualization, C2 in van Wijk's economic model to measure the value of a visualization [van Wijk, 2005]. Thus, increasing the discoverability of interaction techniques may lead to a lower cost of a visualization technique. Moreover, high discoverability may favor the long-term acceptance of a visualization thus play a role towards the **M**aturity of the field expressed in the BRETAM model [Gaines, 1991].

### 7.3.3 *Towards an Infovis Interaction Input Mapping Standard*

In order to improve the discoverability of interactions in Infovis, there is a urgent need of defining a mapping between interactions and input devices. Several standard interaction techniques for Infovis are usually triggered consistently across applications, by convention. For example, in a desktop, mouse and keyboard setup, panning is usually performed by dragging the background of the visualization, zooming using the mouse wheel, and selection by clicking a graphical object in the visualization. Only a few number of interactions are consistent across applications whereas the external consistency of interaction techniques input mapping would accelerate the discoverability of visualizations.

#### 7.3.3.1 *External Consistency*

In the world of industry, Apple provides guidelines[1] for the consistency across Mac applications in terms of visual appearance and design. However, in terms of interaction and user experience, they reformulate many principles from direct manipulation and restrain interaction guidelines to standard menus and icons. Video game controllers are another example illustrating the need for conventions. Moving a character in a first-person shooter game is performed either by using the four arrow keys, either by using w, a, s, and d keys. Pointing targets is performed using the mouse, and shooting usually using the mouse buttons. Moreover, common actions are mapped to the same keys by convention (e. g., *space bar* to jump and *mouse wheel* to change weapon). By using such conventions, video game players transfer their knowledge from one video game, to another, and reduce the amount of features to discover thus accelerate their learning curve. Some commercial products making use of visualization also established conventions. A video editor always contains a timeline on which several actions are performed consistently across applications (e. g., panning, zooming, brushing). Not agreeing on conventions confuses the users and leads to their drop out. For example, Adobe Premiere and Sony Vegas, two video editing software, present different key modifiers to pan and zoom in the timeline, confusing the user. This problem also occurs across operating systems, such as Microsoft Windows and Mac OS that do not map the same hotkeys to the same functions.

Research in HCI may also lead to standards. For example, Fitts' law [Fitts, 1954], certainly the most studied law in HCI (see [Soukoreff and MacKenzie, 2004] for a review of 27 years of Fitts' law research), leaded to the ISO norm 9241-9 for user interfaces.

Conventions may also evolve. For example, at a time Apple decided to inverse the $y$ axis of the trackpad when scrolling. Similarly, video games used to inverse the $y$ axis of the mouse. When such a mistake has been made, developers then have to provide the choice to the users as some of them are now used to a "bad" input mapping. In video games, the transition was long, and most games proposed a setting menu to invert the $y$ axis of the mouse.

Finally, users learn conventions. Despite almost no affordances are available for interacting with smartphones, users quickly become knowledgeable and master the system. Again, consistency across smartphone operating systems helps users transfer their knowledge.

1 https://developer.apple.com/librarY/mac/documentation/UserExperience/Conceptual/AppleHIGuidelines/Intro/Intro.html

#### 7.3.3.2 *Infovis Standard*

In Infovis, only a few conventions exist for very basic interactions. Thus, each Infovis system providing new interactions arbitrarily maps these interactions to the input devices. As a result, Infovis systems are highly externally inconsistent, their discoverability is low, and the learning curve of the user is slow.

Establishing conventions would be a first step towards an interaction input mapping standard for Infovis. First, a standard would steepen the learning curve of users by reducing the amount of discoveries needed and increase transfer of knowledge. Second, it would help visualization designers and student decide which input mapping is appropriate for which existing interaction, thus limit the proliferation of input mappings triggering the same interaction. Third, classifying interaction techniques according to available devices' possibilities would make possible both to identify misused input mappings (e. g., scrolling along one spatial dimension using a two-dimensional mouse pointer), and to explore unused dimensions of input devices (e. g., crossets in Chapter 6). Finally, providing a standard would increase the compatibility within interaction techniques. For example, in Chapter 3, we proposed values zooming and baseline panning by dragging the mouse buttons. However, dragging the mouse is often used for panning and for brushing (e. g., [Javed and Elmqvist, 2010]). Implementing these three interaction techniques in the same visualization would raise compatibility issues and force the user to trigger interaction modes (e. g., using key modifiers). Similarly, complex interaction input mappings that I described in Chapter 2 (e. g., Kronominer, Kinetica, and marking menus for manipulating networks) are valid for a given visualization but not versatile nor compatibles. But what is the standard? which interaction should trigger dragging the mouse on the visualization background? Maybe popular applications such as Tableau [Tableau Software, 2014] have a role to play in establishing and sustaining this standard.

To summarize, establishing an interaction input mapping standard for Infovis would have the following benefits:

1. Increasing Infovis systems' discoverability.
2. Providing implementation guidelines to developers, designers, and students.
3. Identifying unused dimensions of input devices by exploring the design space of possible mappings between input devices and visualizations.
4. Making interaction techniques compatible.

# 8 CONCLUSION

*If you can articulate very precisely what you're seeking, visualization likely isn't your best approach.*

— John T. Stasko [Stasko, 2014]

Visualization is moving to collaborative settings, touch and tangible interfaces, and physical devices. However, the classic desktop is still the default medium today and I believe it will still be for several decades. While it is important to move towards new technologies, a lot remains to be explored to make on-desktop visualization more efficient, more engaging, and more popular. I argue that one key is interaction, as this major component of visualization offers new possibilities provided by the field of HCI and should be considered as equal sibling with representation. This thesis, at the intersection of HCI and Infovis, emphasizes the need for considering the whole spectrum of interaction, encompassing the user's intent (*Why*), the system feedback (*What*), and the actions from the user (*How*). In particular, the *How* is the communication channel between the user and the visualization, thus between the user and the data. As tasks are high level, unknown, and vague, interaction provides the user a way of establishing a dialog between users and data and offers exploration possibilities to facilitate *understanding*, the primary function of visualization.

By opportunistically designing new interactions for visualizations based on recent advances in HCI to go beyond the standard widgets and WIMP interfaces, I showed that visual representations could be unified (Chapter 3), that interaction techniques could be unified (Chapter 5), that new visual representations could be discovered (Chapter 3), and that new interactive instruments could be generated (Chapter 4, Chapter 6).

I addressed several challenges of direct manipulation interfaces, and proposed design guidelines and new challenges for efficient interactions for Infovis such as object of interest-driven interaction design, interaction congruence and versatility, seamless, playful, and engaging interaction, and slow interaction. The gathered lessons and design guidelines are one more step towards a theory of interaction for Infovis.

Based on observations drawn from several projects, I provided perspectives for future research: the need for comparing interactive systems, the role and factors of interaction techniques' discoverability, and the need for an Infovis interaction mapping standard.

Finally, opportunistic interaction and pragmatism are an alternative strategy to the formalization of a theory of interaction for Infovis—that may never be established. More than an alternative, this strategy is a catalyst for creativity as no formalization restrains innovations.

# BIBLIOGRAPHY


Johnny Accot and Shumin Zhai. More than dotting the i's — foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 73–80, New York, NY, USA, 2002. ACM. ISBN 1-58113-453-3. doi: 10.1145/503376.503390. URL http://doi.acm.org/10.1145/503376.503390. (Cited on pages 127, 130, 146, and 147.)

Anand Agarawala and Ravin Balakrishnan. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 1283–1292, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124965. URL http://doi.acm.org/10.1145/1124772.1124965. (Cited on page 158.)

Christopher Ahlberg and Ben Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 313–317, New York, NY, USA, 1994. ACM. ISBN 0-89791-650-6. doi: 10.1145/191666.191775. URL http://doi.acm.org/10.1145/191666.191775. (Cited on pages 19 and 25.)

Christopher Ahlberg, Christopher Williamson, and Ben Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. In *Proc. CHI'92*, pages 619–626. ACM, 1992. (Cited on pages 25 and 147.)

Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, 2011. ISBN 978-0-85729-078-6. URL http://dx.doi.org/10.1007/978-0-85729-079-3. (Cited on pages 41 and 50.)

Robert Amar, James Eagan, and John Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 15–, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7803-9464-x. doi: 10.1109/INFOVIS.2005.24. URL http://dx.doi.org/10.1109/INFOVIS.2005.24. (Cited on pages 4, 23, 24, and 70.)

John Robert Anderson. *Learning and memory: An Integrated Approach*, volume 86. Wiley, New York, 2 edition, 2000. ISBN 9780471249252. (Cited on page 85.)

Gennady L. Andrienko and Natalia V. Andrienko. Interactive maps for visual data exploration. *International Journal of Geographical Information Science*, 13(4):355–374, 1999. doi: doi:10.1080/136588199241247. (Cited on page 30.)

Natalia Andrienko and Gennady Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 3540259945. (Cited on pages vi, 43, 50, 51, 70, 71, and 90.)

Ian G. Angus and Henry A. Sowizral. Embedding the 2d interaction metaphor in a real 3d virtual environment. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems II*, volume 2409, pages 282–293, 1995. doi: 10.1117/12.205875. URL http://dx.doi.org/10.1117/12.205875. (Cited on page 20.)

Alissa N. Antle, Greg Corness, and Milena Droumeva. What the body knows: Exploring the benefits of embodied metaphors in hybrid physical digital environments. *Interacting with Computers*, 21(1–2):66 – 75, 2009. ISSN 0953-5438. doi: http://dx.doi.org/10.1016/j.intcom.2008.10.005. URL http://www.sciencedirect.com/science/article/pii/S0953543808000647. Special issue: Enactive Interfaces Physicality & Interaction. (Cited on page 158.)

Georg Apitz and François Guimbretière. Crossy: A crossing-based drawing application. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 3–12, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: 10.1145/1029632.1029635. URL http://doi.acm.org/10.1145/1029632.1029635. (Cited on pages 127 and 147.)

Caroline Appert and Jean-Daniel Fekete. Orthozoom scroller: 1d multi-scale navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 21–30, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124776. URL http://doi.acm.org/10.1145/1124772.1124776. (Cited on page 127.)

ARTICQUE. Cartes et données, 2013. Available at www.articque.com/solutions/cartes-et-donnees/. (Cited on page 122.)

Isaac Asimov. *Second Foundation*. Book Club edition. Gnome Press, 1953. URL http://books.google.fr/books?id=YT4GAQAAIAAJ. (Cited on page xix.)

Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. Visualizing dynamic networks with matrix cubes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 877–886, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557010. URL http://doi.acm.org/10.1145/2556288.2557010. (Cited on page 86.)

Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. The rockin'mouse: Integral 3d manipulation on a plane. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 311–318, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258778. URL http://doi.acm.org/10.1145/258549.258778. (Cited on page 20.)

Ziv Bar-Joseph, David K. Gifford, and Tommi S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(suppl 1):S22–S29, 2001. doi: 10.1093/bioinformatics/17.suppl_1.S22. URL http://bioinformatics.oxfordjournals.org/content/17/suppl_1/S22.abstract. (Cited on pages 120 and 135.)

Thomas Baudel. From information visualization to direct manipulation: Extending a generic visualization framework for the interactive editing of large datasets. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 67–76, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1. doi: 10.1145/1166253.1166265. URL http://doi.acm.org/10.1145/1166253.1166265. (Cited on page 7.)

Patrick Baudisch. Don't click, paint! using toggle maps to manipulate sets of toggle switches. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 65–66, New York, NY, USA, 1998. ACM. ISBN 1-58113-034-1. doi: 10.1145/288392.288574. URL http://doi.acm.org/10.1145/288392.288574. (Cited on pages 127 and 147.)

Patrick Baudisch, Edward Cutrell, Dan Robbins, Mary Czerwinski, Peter Tandler, Benjamin B. Bederson, Alex Zierlinger, et al. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *Proceedings of the International Conference on Human-Computer Interaction*, volume 3 of *INTERACT '03*, pages 57–64, 2003. (Cited on pages 91, 92, and 95.)

Dominikus Baur, Bongshin Lee, and Sheelagh Carpendale. Touchwave: Kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 255–264, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396675. URL http://doi.acm.org/10.1145/2396636.2396675. (Cited on page 24.)

Vivien Beattie and Michael John Jones. The impact of graph slope on rate of change judgments in corporate reports. *Abacus*, 38(2):177–199, 2002. (Cited on page 44.)

Michel Beaudouin-Lafon. Interaction instrumentale: de la manipulation directe à la réalité augmentée. In *Proceedings of the 9th days on Interaction Homme-Machine*, IHM '97. Cépaduès-Editions, Sep 1997. (Cited on pages 6, 14, and 156.)

Michel Beaudouin-Lafon. Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 446–453, New York, NY, USA, 2000. ACM. ISBN 1-58113-216-6. doi: 10.1145/332040.332473. URL http://doi.acm.org/10.1145/332040.332473. (Cited on pages 5, 6, 14, 15, 31, 122, 127, 145, 155, and 156.)

Michel Beaudouin-Lafon and Henry Michael Lassen. The architecture and implementation of cpn2000, a post-wimp graphical application. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, UIST '00, pages 181–190, New York, NY, USA, 2000. ACM. ISBN 1-58113-212-3. doi: 10.1145/354401.354761. URL http://doi.acm.org/10.1145/354401.354761. (Cited on page 14.)

Michel Beaudouin-Lafon and Wendy E. Mackay. Reification, polymorphism and reuse: Three principles for designing visual interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, pages 102–109, New York, NY, USA, 2000. ACM. ISBN 1-58113-252-2. doi: 10.1145/345513.345267. URL http://doi.acm.org/10.1145/345513.345267. (Cited on pages 6 and 14.)

Michel Beaudouin-Lafon, Wendy E. Mackay, Peter Andersen, Paul Janecek, Mads Jensen, Henry Michael Lassen, Kasper Lund, Kjeld Høyer Mortensen, Stephanie Munck, Anne V. Ratzer, Katrine Ravn, Søren Christensen, and Kurt Jensen. Cpn/tools: A post-wimp interface for editing and simulating coloured petri nets. In *Proceedings of the 22Nd International Conference on Application and Theory of Petri Nets*, ICATPN '01, pages 71–80, London, UK, UK, 2001. Springer-Verlag. ISBN 3-540-42252-8. URL http://dl.acm.org/citation.cfm?id=647747.734225. (Cited on page 14.)

Benjamin B. Bederson. Interfaces for staying in the flow. *Ubiquity*, 2004 (September):1–1, September 2004. ISSN 1530-2180. doi: 10.1145/1074068.1074069. URL http://doi.acm.org/10.1145/1074068.1074069. (Cited on pages 7, 15, 66, 69, 74, 83, 85, 87, 88, and 153.)

Benjamin B. Bederson and Angela Boltman. Does animation help users build mental maps of spatial information? In *Proceedings of the 1999 IEEE Symposium on Information Visualization*, INFOVIS '99, pages 28–, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0431-0. URL http://dl.acm.org/citation.cfm?id=857189.857656. (Cited on page 104.)

Benjamin B. Bederson and James D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, pages 17–26, New York, NY, USA, 1994. ACM. ISBN 0-89791-657-3. doi: 10.1145/192426.192435. URL http://doi.acm.org/10.1145/192426.192435. (Cited on page 45.)

William H. Benson and Bernard Kitous. Interactive analysis and display of tabular data. *SIGGRAPH Comput. Graph.*, 11(2):48–53, July 1977. ISSN 0097-8930. doi: 10.1145/965141.563869. URL http://doi.acm.org/10.1145/965141.563869. (Cited on pages 123 and 135.)

Jacques Bertin. *Sémiologie graphique : Les diagrammes-Les réseaux-Les cartes*. Mouton/Gauthier-Villars, 1967. (Cited on pages 2, 114, 116, and 119.)

Jacques Bertin. Graphique et mathématique: Généralisation du traitement graphique de l'information. *Annales. Économies, Sociétés, Civilisations*, 24 (1):70–101, 1969. ISSN 0395-2649. doi: 10.3406/ahess.1969.422034. (Cited on page 116.)

Jacques Bertin. *La graphique et le traitement graphique de l'information*. Nouvelle bibliothèque scientifique. Flammarion, 1975. ISBN 2-08-211112-1. (Cited on pages 26, 114, 116, 117, 118, 120, 126, 134, 136, 143, 145, 146, and 152.)

Jacques Bertin. Cahier des charges pour un mini-ordinateur "graphique". Actes du second Colloque de micro-info-graphique de Rouen (Unpublished). Available at `http://www.aviz.fr/wiki/uploads/Bertifier/bertin-1982-cahier.pdf`, 1982. (Cited on pages 122 and 135.)

Jacques Bertin. *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin Press, 1983. ISBN 0299090604. (Cited on pages 41, 43, and 108.)

Jacques Bertin. *Sémiologie graphique : Les diagrammes-Les réseaux-Les cartes*. Éditions de l'Ecole des Hautes Etudes en Sciences, Paris, France, les réimpressions edition, 1999. (Cited on pages 116 and 119.)

Jacques Bertin and Jean-Hugues. Chauchat. Logiciel amado (analyse graphique d'une matrice de données). *CISIA, France*, 1994. (Cited on pages 26, 114, 123, and 133.)

Jaques Bertin. *Graphics and Graphic Information Processing*. De Gruyter, Berlin, 1981. ISBN 9783110088687. Translation: William J. Berg, Paul Scott. (Cited on pages 114, 116, 117, and 119.)

Alan F. Blackwell. The reification of metaphor as a design tool. *ACM Trans. Comput.-Hum. Interact.*, 13(4):490–530, December 2006. ISSN 1073-0516. doi: 10.1145/1188816.1188820. URL `http://doi.acm.org/10.1145/1188816.1188820`. (Cited on page 19.)

Susanne Bødker. A human activity approach to user interfaces. *Hum.-Comput. Interact.*, 4(3):171–195, September 1989. ISSN 0737-0024. doi: 10.1207/s15327051hci0403_1. URL `http://dx.doi.org/10.1207/s15327051hci0403_1`. (Cited on page 14.)

Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12): 2301–2309, December 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2011.185. URL `http://dx.doi.org/10.1109/TVCG.2011.185`. (Cited on pages 6, 77, and 96.)

Mike Bostock. The wealth & health of nations, 2012. `http://bost.ocks.org/mike/nations/`. (Cited on page 105.)

Jeremy Boy, Ronald A. Rensink, Enrico Bertini, and Jean-Daniel Fekete. A principled way of assessing visualization literacy. *Visualization and Computer Graphics, IEEE Transactions on*, 2014. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2346984. (Cited on pages ix and 16.)

U. Brandes and S.R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. *IEEE Transactions on Visualization and Computer Graphics*, pages 145–151, 2002. ISSN 1522-404X. doi: 10.1109/INFVIS.2002.1173160. (Cited on pages 91 and 95.)

Matthew Brehmer and Tamara Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, December 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.124. URL http://dx.doi.org/10.1109/TVCG.2013.124. (Cited on page 4.)

John Brosz, Miguel A. Nacenta, Richard Pusch, Sheelagh Carpendale, and Christophe Hurter. Transmogrification: Causal manipulation of visualizations. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 97–106, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2502046. URL http://doi.acm.org/10.1145/2501988.2502046. (Cited on page 91.)

Jeffrey Browne, Bongshin Lee, Sheelagh Carpendale, Nathalie Riche, and Timothy Sherwood. Data analysis on interactive whiteboards through sketch-based interaction. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pages 154–157, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0871-7. doi: 10.1145/2076354.2076383. URL http://doi.acm.org/10.1145/2076354.2076383. (Cited on page 24.)

Andreas Buja, Dianne Cook, and Deborah F. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5:78–99, 1996. (Cited on page 23.)

Lee Byron and Martin Wattenberg. Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6): 1245–1252, November 2008. ISSN 1077-2626. doi: 10.1109/TVCG.2008.166. URL http://dx.doi.org/10.1109/TVCG.2008.166. (Cited on page 41.)

Gilles Caraux and Sylvie Pinloche. PermutMatrix: a graphical environment to arrange gene expression profiles in optimal linear order. *Bioinformatics*, 21(7):1280–1281, April 2005. ISSN 1460-2059. doi: 10.1093/bioinformatics/bti141. URL http://dx.doi.org/10.1093/bioinformatics/bti141. (Cited on pages 121 and 123.)

Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1983. ISBN 0898592437. (Cited on page 155.)

Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999. ISBN 1-55860-533-9. (Cited on pages 2, 3, 4, 23, and 116.)

Sheelagh Carpendale. *A Framework for Elastic Presentation Space*. PhD thesis, Simon Fraser University, Burnaby, BC, Canada, Canada, 1999. AAINQ51848. (Cited on page 23.)

J. M. Carroll. Learning, using and designing filenames and command paradigms. *Behaviour & Information Technology*, 1(4):327–346, 1982. (Cited on pages 17 and 18.)

Stephen M. Casner. Task-analytic approach to the automated design of graphic presentations. *ACM Trans. Graph.*, 10(2):111–151, April 1991. ISSN 0730-0301. doi: 10.1145/108360.108361. URL http://doi.acm.org/10.1145/108360.108361. (Cited on page 86.)

Fanny Chevalier, Pierre Dragicevic, Anastasia Bezerianos, and Jean-Daniel Fekete. Using text animated transitions to support navigation in document histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 683–692, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753427. URL http://doi.acm.org/10.1145/1753326.1753427. (Cited on page 91.)

Fanny Chevalier, Pierre Dragicevic, and Christophe Hurter. Histomages: Fully synchronized views for image editing. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 281–286, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1580-7. doi: 10.1145/2380116.2380152. URL http://doi.acm.org/10.1145/2380116.2380152. (Cited on page 35.)

Ed H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 69–, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0804-9. URL http://dl.acm.org/citation.cfm?id=857190.857691. (Cited on page 4.)

Ed H. Chi and John Riedl. An operator interaction framework for visualization systems. In *Proceedings of the 1998 IEEE Symposium on Information Visualization*, INFOVIS '98, pages 63–70, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 0-8186-9093-3. URL http://dl.acm.org/citation.cfm?id=647341.721078. (Cited on pages 4 and 23.)

Mei C. Chuah and Steven F. Roth. On the semantics of interactive visualizations. In *Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, INFOVIS '96, pages 29–, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7668-X. URL http://dl.acm.org/citation.cfm?id=857187.857620. (Cited on page 23.)

Wim Claasen, Edwin Bos, and Carla Huls. The pooh way in human-computer interaction: towards multimodal interfaces. SPIN MC Research report #5, 1990. (Cited on page 18.)

Andy Clark. *Being There: Putting Brain, Body, and World Together Again*. MIT Press, Cambridge, MA, 1997. ISBN 9780262260527. URL http://books.google.fr/books?id=i03NKy0ml1gC. (Cited on page 37.)

William S. Cleveland. *The Elements of Graphing Data*. Wadsworth Publ. Co., Belmont, CA, USA, 1985. ISBN 0-534-03730-5. (Cited on page 41.)

William S. Cleveland. *Visualizing Data*. Hobart Press, 1993. ISBN 0963488406. (Cited on page 40.)

William S. Cleveland and Robert McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, pages 531–554, 1984. (Cited on page 2.)

William S. Cleveland and Robert McGill. Graphical Perception: The Visual Decoding of Quantitative Information on Graphical Displays of Data. *Journal of the Royal Statistical Society*, 150(3):192–229, 1987. ISSN 00359238. (Cited on page 43.)

Andy Cockburn and Bruce McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 203–210, New York, NY, USA, 2002. ACM. ISBN 1-58113-453-3. doi: 10.1145/503376.503413. URL http://doi.acm.org/10.1145/503376.503413. (Cited on page 24.)

Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, January 2009. ISSN 0360-0300. doi: 10.1145/1456650.1456652. URL http://doi.acm.org/10.1145/1456650.1456652. (Cited on page 42.)

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. L. Erlbaum Associates, 2 edition, July 1988. ISBN 9780805802832. (Cited on pages 53 and 54.)

Brookshire D. Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. Three-dimensional widgets. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, I3D '92, pages 183–188, New York, NY, USA, 1992. ACM. ISBN 0-89791-467-8. doi: 10.1145/147156.147199. URL http://doi.acm.org/10.1145/147156.147199. (Cited on page 20.)

Shanghai Ranking Consultancy. Academic ranking of world universities 2013, 2013. http://www.shanghairanking.com/. (Cited on pages 84 and 98.)

Kenneth Cox, Rebecca E. Grinter, and David Mantilla. Using dialog and context in a speech-based interface for an information visualization environment. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, pages 274–275, New York, NY, USA, 2000. ACM. ISBN 1-58113-252-2. doi: 10.1145/345513.345342. URL http://doi.acm.org/10.1145/345513.345342. (Cited on page 24.)

M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper Collins, New York, NY, 1991. ISBN 0060920432. (Cited on pages 7 and 15.)

Raimund Dachselt and Anett Hübner. Virtual environments: Three-dimensional menus: A survey and taxonomy. *Comput. Graph.*, 31(1):53–65, January 2007. ISSN 0097-8493. doi: 10.1016/j.cag.2006.09.006. URL http://dx.doi.org/10.1016/j.cag.2006.09.006. (Cited on page 20.)

Antoine de Falguerolles, Felix Friedrich, and Günther Sawitzki. A tribute to J. Bertin's graphical data analysis. In *Advances in Statistical Software 6*, pages 11–20. Lucius & Lucius, 1997. ISBN 3-8282-0032-X. (Cited on page 123.)

Josep Díaz, Jordi Petit, and Maria Serna. A survey of graph layout problems. *ACM Comput. Surv.*, 34(3):313–356, September 2002. ISSN 0360-0300. doi: 10.1145/568522.568523. URL http://doi.acm.org/10.1145/568522.568523. (Cited on page 120.)

Alan Dix and Geoffrey Ellis. Starting simple: Adding value to static visualisation through simple interaction. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '98, pages 124–134, New York, NY, USA, 1998. ACM. doi: 10.1145/948496.948514. URL http://doi.acm.org/10.1145/948496.948514. (Cited on pages ix, 4, 19, 23, 39, 111, 113, 116, and 153.)

Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 237–246, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357096. URL http://doi.acm.org/10.1145/1357054.1357096. (Cited on pages 16, 91, 92, 95, 103, 104, and 152.)

Pierre Dragicevic, Anastasia Bezerianos, Waqas Javed, Niklas Elmqvist, and Jean-Daniel Fekete. Temporal distortion for animated transitions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2009–2018, New York, NY, USA, 2011a. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979233. URL http://doi.acm.org/10.1145/1978942.1979233. (Cited on pages 95, 102, and 103.)

Pierre Dragicevic, Stéphane Huot, and Fanny Chevalier. Gliimpse: Animating from markup code to rendered documents and vice versa. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 257–262, New York, NY, USA, 2011b. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047229. URL http://doi.acm.org/10.1145/2047196.2047229. (Cited on pages 91 and 103.)

Stephen G. Eick. Data visualization sliders. In *ACM Symposium on User Interface Software and Technology*, pages 119–120, 1994. (Cited on page 30.)

Michael Eisenberg. The thin glass line: Designing interfaces to algorithms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pages 181–188, New York, NY, USA, 1996. ACM. ISBN 0-89791-777-4. doi: 10.1145/238386.238474. URL http://doi.acm.org/10.1145/238386.238474. (Cited on page 21.)

Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, Nov 2008a. ISSN 1077-2626. doi: 10.1109/TVCG.2008.153. (Cited on pages 32, 86, and 91.)

Niklas Elmqvist, Nathalie Henry Riche, Yann Riche, and Jean-Daniel Fekete. Melange: Space folding for multi-focus interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1333–1342, New York, NY, USA, 2008b. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357263. URL http://doi.acm.org/10.1145/1357054.1357263. (Cited on page 69.)

Niklas Elmqvist, Andrew Vande Moere, Hans-Christer Jetter, Daniel Cernea, Harald Reiterer, and T.J. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, 2011. (Cited on pages 6 and 15.)

EVS. European values study 1981–2008, longitudinal data file, 2011. (Cited on page 124.)

Christine Faulkner. *The Essence of Human-computer Interaction*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998. ISBN 0-13-751975-3. (Cited on page 19.)

Jean-Daniel Fekete. The infovis toolkit. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 167–174, 2004. doi: 10.1109/INFVIS.2004.64. (Cited on page 127.)

Jean-Daniel Fekete and Catherine Plaisant. Excentric labeling: Dynamic neighborhood labeling for data visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 512–519, New York, NY, USA, 1999. ACM. ISBN 0-201-48559-1. doi: 10.1145/302979.303148. URL http://doi.acm.org/10.1145/302979.303148. (Cited on page 108.)

Jean-Daniel Fekete, Jarke J. van Wijk, John T. Stasko, and Chris North. Information visualization. chapter The Value of Information Visualization, pages 1–18. Springer-Verlag, Berlin, Heidelberg, 2008. ISBN 978-3-540-70955-8. doi: 10.1007/978-3-540-70956-5_1. URL http://dx.doi.org/10.1007/978-3-540-70956-5_1. (Cited on pages 4, 37, and 118.)

Stephen Few. Time on the horizon. available online at http://www.perceptualedge.com/articles/visual_business_intelligence/time_on_the_horizon.pdf, 2008. (Cited on pages 41, 42, and 43.)

Stephen Few. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press, USA, 1st edition, 2009. ISBN 0970601980, 9780970601988. (Cited on page 1.)

Stephen Few. Visweek 2011 – award-worthy visualization research. http://www.perceptualedge.com/blog/?p=1095. Online, accessed 2014-07-06., 2011. (Cited on pages 20 and 35.)

Stephen Few. The slow data movement: My hope for 2013. http://www.perceptualedge.com/blog/?p=1460. Online, accessed 2014-07-06., 2013. (Cited on pages 118 and 154.)

P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental PSychology*, 74: 381–391, 1954. (Cited on page 161.)

Jean-Michel Fras, Roberto Gimeno, and Pierre-Yves Vicens. *Matrix: programme de traitement matriciel en Basic TO7*. CNDP, 1984. ISBN 9782240000361. (Cited on page 122.)

Mathias Frisch, Jens Heydekorn, and Raimund Dachselt. Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 149–156, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731933. URL http://doi.acm.org/10.1145/1731903.1731933. (Cited on page 24.)

David M. Frohlich. The history and future of direct manipulation. *Behaviour & Information Technology*, 12(6):315–329, 1993. ISSN 0144-929X. doi: 10.1080/01449299308924396. (Cited on pages 18 and 153.)

David M. Frohlich. Direct manipulation and other lessons. In *Handbook of human-computer interaction (2nd ed)*, pages 463–488. Elsevier, 1997. (Cited on pages 16 and 18.)

Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, Fabian Fischer, and Enrico Bertini. The influence of contour on similarity perception of star glyphs. *TVCG*, 2014. To appear. (Cited on page 139.)

Brian R. Gaines. Modeling and forecasting the information sciences. *Inf. Sci.*, 57-58:3–22, September 1991. ISSN 0020-0255. URL http://dl.acm.org/citation.cfm?id=106027.106028. (Cited on pages 37 and 160.)

M. Gardet and Y. Irid. Archives du laboratoire de cartographie, inventaire, 2012. (Cited on page 115.)

M. Gattis. Reading pictures: constraints on mapping conceptual and spatial schemas. In *Spatial schemas and abstract thought*, page 223–245. MIT Press, Cambridge, MA, 2001. (Cited on page 16.)

M. Gattis and K. J. Holyoak. Mapping conceptual to spatial relations in visual reasoning. *Journal of experimental psychology. Learning, memory, and cognition*, 22(1):231–239, January 1996. ISSN 0278-7393. (Cited on page 16.)

Don Gentner and Jakob Nielsen. The anti-mac interface. *Commun. ACM*, 39 (8):70–82, August 1996. ISSN 0001-0782. doi: 10.1145/232014.232032. URL http://doi.acm.org/10.1145/232014.232032. (Cited on page 19.)

Sohaib Ghani, Nathalie Henry Riche, and Niklas Elmqvist. Dynamic insets for context-aware graph navigation. In *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'11, pages 861–870, Aire-la-Ville, Switzerland, Switzerland, 2011. Eurographics Association. doi: 10.1111/j.1467-8659.2011.01935.x. URL http://dx.doi.org/10.1111/j.1467-8659.2011.01935.x. (Cited on page 92.)

James Jerome Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. ISBN 9780395270493. (Cited on pages 6 and 19.)

Wilfried A. Gibson. A simple procedure for rearranging matrices. *Psychometrika*, 18(2):111–113, 1953. ISSN 0033-3123. doi: 10.1007/BF02289000. URL http://dx.doi.org/10.1007/BF02289000. (Cited on page 116.)

Roberto Gimeno and Marie-Françoise Durand. Mise en place et utilisation d'un logiciel de cartographie et traitement des données dans l'enseignement de la géographie. *Revue de géographie de Lyon*, 63(2):64–68, 1988. ISSN 0035-113X. doi: 10.3406/geoca.1988.3370. URL http://www.persee.fr/web/revues/home/prescript/article/geoca_0035-113x_1988_num_63_2_3370. (Cited on pages 26 and 122.)

Yves Girault and Thérèse Rukingama. Evaluation de l'exploitation d'une visite au zoo par le traitement graphique. 1988. doi: 10.4267/2042/9170. (Cited on page 122.)

Google Finance. Google finance. http://www.google.com/finance, 2012. (Cited on page 49.)

R. Gosselin. T_alk, 2008. Available at http://rg75.free.fr/hyperespace.T_AlK.fr.htm. (Cited on page 123.)

Lars Grammel, Chris Bennett, Melanie Tory, and Margaret-Anne Storey. A Survey of Visualization Construction User Interfaces. pages 19–23, Leipzig, Germany, 2013. Eurographics. ISBN 978-3-905673-99-9. doi: 10.2312/PE.EuroVisShort.EuroVisShort2013.019-023. URL http://diglib.eg.org/EG/DL/PE/EuroVisShort/EuroVisShort2013/019-023.pdf. (Cited on page 15.)

Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, December 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.173. URL http://dx.doi.org/10.1109/TVCG.2013.173. (Cited on pages 66, 68, and 89.)

Tera Marie Green, William Ribarsky, and Brian Fisher. Visual analytics for complex concepts using a human cognition model. In *Visual Analytics Science and Technology, 2008. VAST '08. IEEE Symposium on*, pages 91–98, Oct 2008. doi: 10.1109/VAST.2008.4677361. (Cited on page 15.)

Thomas R. G. Green and Marian Petre. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing*, 7(2):131–174, 1996. (Cited on page 126.)

Bert F Green Jr. The computer revolution in psychometrics. *Psychometrika*, 31(4):437–445, 1966. ISSN 0033-3123. doi: 10.1007/BF02289515. URL http://dx.doi.org/10.1007/BF02289515. (Cited on page 118.)

Saul Greenberg, Nicolai Marquardt, Till Ballendat, Rob Diaz-Marino, and Miaosen Wang. Proxemic interactions: The new ubicomp? *interactions*, 18 (1):42–50, January 2011. ISSN 1072-5520. doi: 10.1145/1897239.1897250. URL http://doi.acm.org/10.1145/1897239.1897250. (Cited on pages 5 and 14.)

Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. Multi-finger gestural interaction with 3d volumetric displays. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 61–70, New York, NY, USA, 2004. ACM. ISBN 1-58113-957-8. doi: 10.1145/1029632.1029644. URL http://doi.acm.org/10.1145/1029632.1029644. (Cited on page 20.)

Jonathan Grudin. The case against user interface consistency. *Commun. ACM*, 32(10):1164–1173, October 1989. ISSN 0001-0782. doi: 10.1145/67933.67934. URL http://doi.acm.org/10.1145/67933.67934. (Cited on pages ix, 6, 17, 18, 19, 20, and 35.)

Louis Guttman. What is not what in statistics. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 26(2):81–107, 1977. ISSN 00390526. (Cited on page 118.)

Joann T. Hackos and Janice C. Redish. *User and Task Analysis for Interface Design*. John Wiley & Sons, Inc., New York, NY, USA, 1998. ISBN 0-471-17831-4. (Cited on page 19.)

Frank Halasz and Thomas P. Moran. Analogy considered harmful. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems*, CHI '82, pages 383–386, New York, NY, USA, 1982. ACM. doi: 10.1145/800049.801816. URL http://doi.acm.org/10.1145/800049.801816. (Cited on pages 17, 19, and 20.)

Granville S. Hall and Edward B. Titchener. *The American Journal of Psychology*. University of Illinois Press, 1903. (Cited on page 159.)

H. Rex Hartson, Antonio C. Siochi, and Deborah Hix. The uan: A user-oriented representation for direct manipulation interface designs. *ACM Trans. Inf. Syst.*, 8(3):181–203, July 1990. ISSN 1046-8188. doi: 10.1145/98188.98191. URL http://doi.acm.org/10.1145/98188.98191. (Cited on page 155.)

Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular brushing of extended parallel coordinates. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, INFOVIS '02, pages 127–, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1751-X. URL http://dl.acm.org/citation.cfm?id=857191.857742. (Cited on page 27.)

Jeffrey Heer and Maneesh Agrawala. Multi-scale banking to 45 degrees. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):701–708, September 2006. ISSN 1077-2626. doi: 10.1109/TVCG.2006.163. URL http://dx.doi.org/10.1109/TVCG.2006.163. (Cited on page 43.)

Jeffrey Heer and George Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6): 1240–1247, November 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70539. URL http://dx.doi.org/10.1109/TVCG.2007.70539. (Cited on pages 76 and 91.)

Jeffrey Heer and Ben Shneiderman. Interactive dynamics for visual analysis. *Commun. ACM*, 55(4):45–54, April 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133821. URL http://doi.acm.org/10.1145/2133806.2133821. (Cited on pages 4 and 24.)

Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1303–1312, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518897. URL http://doi.acm.org/10.1145/1518701.1518897. (Cited on pages 40, 42, 43, 44, 46, 49, 50, 51, 58, and 59.)

Martin Helander. *Handbook of Human-Computer Interaction*. North-Holland, 1988. ISBN 9780444705365. (Cited on page 20.)

Nathalie Henry Riche and Jean-Daniel Fekete. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):677–684, Sept 2006. ISSN 1077-2626. doi: 10.1109/TVCG.2006.160. (Cited on pages 114, 119, 121, 122, and 133.)

Nathalie Henry Riche, Bongshin Lee, and Catherine Plaisant. Understanding interactive legends: A comparative evaluation with standard widgets. In *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'10, pages 1193–1202, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics. doi: 10.1111/j.1467-8659.2009.01678.x. URL http://dx.doi.org/10.1111/j.1467-8659.2009.01678.x. (Cited on pages 29 and 30.)

S. Hill. *Practical Introduction to the Human Computer Interface in a Semester.* ... in a semester. DP Publications, 1995. ISBN 9781858051192. URL http://books.google.fr/books?id=nCzEAAAACAAJ. (Cited on page 19.)

Hans Hinterberger. The visulab®: an instrument for interactive, comparative visualization. 2010. (Cited on pages 121 and 123.)

Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, March 2004. ISSN 1473-8716. doi: 10.1145/993176.993177. URL http://dx.doi.org/10.1145/993176.993177. (Cited on page 40.)

Christian Holz and Steven K. Feiner. Relaxed selection techniques for querying time-series graphs. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pages 213–222, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-745-5. doi: 10.1145/1622176.1622217. URL http://doi.acm.org/10.1145/1622176.1622217. (Cited on page 24.)

Simo Hosio, Jorge Goncalves, and Vassilis Kostakos. Application discoverability on multipurpose public displays: Popularity comes at a price. In *Proceedings of the 2Nd ACM International Symposium on Pervasive Displays*, PerDis '13, pages 31–36, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2096-2. doi: 10.1145/2491568.2491576. URL http://doi.acm.org/10.1145/2491568.2491576. (Cited on page 158.)

Scott E. Hudson and John T. Stasko. Animation support in a user interface toolkit: Flexible, robust, and reusable abstractions. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*, UIST '93, pages 57–67, New York, NY, USA, 1993. ACM. ISBN 0-89791-628-X. doi: 10.1145/168642.168648. URL http://doi.acm.org/10.1145/168642.168648. (Cited on page 91.)

Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, December 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.227. URL http://dx.doi.org/10.1109/TVCG.2013.227. (Cited on pages 91 and 95.)

Samuel Huron, Yvonne Jansen, and Sheelagh Carpendale. Constructing visual representations: Investigating the use of tangible tokens. *TVCG*, 2014. To appear. (Cited on pages 139 and 142.)

Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. Direct manipulation interfaces. *Hum.-Comput. Interact.*, 1(4):311–338, December 1985. ISSN 0737-0024. doi: 10.1207/s15327051hci0104_2. URL http://dx.doi.org/10.1207/s15327051hci0104_2. (Cited on pages 5, 13, 16, 17, 18, 19, 36, and 159.)

Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985. ISSN 0178-2789. doi: 10.1007/BF01898350. URL http://dx.doi.org/10.1007/BF01898350. (Cited on page 27.)

Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st Conference on Visualization '90*, VIS '90, pages 361–378, Los Alamitos, CA, USA, 1990.

IEEE Computer Society Press. ISBN 0-8186-2083-8. URL http://dl.acm.org/citation.cfm?id=949531.949588. (Cited on page 27.)

Pourang Irani, Carl Gutwin, and Xing Dong Yang. Improving selection of off-screen targets with hopping. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 299–308, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124818. URL http://doi.acm.org/10.1145/1124772.1124818. (Cited on page 92.)

Petra Isenberg, Uta Hinrichs, Mark Hancock, and Sheelagh Carpendale. Digital tables for collaborative information exploration. In Christian Müller-Tomfelde, editor, *Tabletops - Horizontal Interactive Displays*, Human-Computer Interaction Series, pages 387–405. Springer London, 2010. ISBN 978-1-84996-112-7. doi: 10.1007/978-1-84996-113-4_16. URL http://dx.doi.org/10.1007/978-1-84996-113-4_16. (Cited on page 24.)

Petra Isenberg, Anastasia Bezerianos, Pierre Dragicevic, and Jean-Daniel Fekete. A study on dual-scale data charts. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2469–2478, December 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2011.160. URL http://dx.doi.org/10.1109/TVCG.2011.160. (Cited on pages 42 and 45.)

Edward W. Ishak and Steven K. Feiner. Content-aware scrolling. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 155–158, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1. doi: 10.1145/1166253.1166277. URL http://doi.acm.org/10.1145/1166253.1166277. (Cited on page 92.)

Robert J. K. Jacob, Linda E. Sibert, Daniel C. McFarlane, and M. Preston Mullen, Jr. Integrality and separability of input devices. *ACM Trans. Comput.-Hum. Interact.*, 1(1):3–26, March 1994. ISSN 1073-0516. doi: 10.1145/174630.174631. URL http://doi.acm.org/10.1145/174630.174631. (Cited on page 15.)

Robert J. K. Jacob, Hiroshi Ishii, Gian Pangaro, and James Patten. A tangible interface for organizing information using a grid. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 339–346, New York, NY, USA, 2002. ACM. ISBN 1-58113-453-3. doi: 10.1145/503376.503437. URL http://doi.acm.org/10.1145/503376.503437. (Cited on page 24.)

Jacek Jankowski and Martin Hachet. A Survey of Interaction Techniques for Interactive 3D Environments. In *Eurographics 2013 - STAR*, pages 65–93, Girona, Spain, 2012. Eurographics Association. doi: 10.2312/conf/EG2013/stars/065-093. (Cited on pages 20 and 21.)

Yvonne Jansen. *Physical and tangible information visualization*. PhD thesis, Université Paris Sud - Paris XI, March 2014. URL http://tel.archives-ouvertes.fr/tel-00983501. (Cited on page 24.)

Yvonne Jansen and Pierre Dragicevic. An interaction model for visualizations beyond the desktop. *IEEE TVCG*, 19(12):2396–2405, 2013. (Cited on pages 4, 6, 23, 32, 36, and 151.)

Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. Evaluating the efficiency of physical visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2593–2602, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2481359. URL http://doi.acm.org/10.1145/2470654.2481359. (Cited on page 24.)

Waqas Javed and Niklas Elmqvist. Stack zooming for multi-focus interaction in time-series data visualization. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 33–40. IEEE, March 2010. doi: 10.1109/PACIFICVIS.2010.5429613. (Cited on pages 42 and 162.)

Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–934, Nov 2010. ISSN 1077-2626. doi: 10.1109/TVCG.2010.162. (Cited on pages 40, 41, 42, 43, 44, 50, 51, 52, 57, and 59.)

Morgan Jennings. Theory and models for creating engaging and immersive ecommerce websites. In *Proceedings of the 2000 ACM SIGCPR Conference on Computer Personnel Research*, SIGCPR '00, pages 77–85, New York, NY, USA, 2000. ACM. ISBN 1-58113-228-X. doi: 10.1145/333334.333358. URL http://doi.acm.org/10.1145/333334.333358. (Cited on page 153.)

Hans-Christian Jetter, Jens Gerken, Michael Zöllner, Harald Reiterer, and Natasa Milic-Frayling. Materializing the query with facet-streams: A hybrid surface for collaborative search on tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3013–3022, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979390. URL http://doi.acm.org/10.1145/1978942.1979390. (Cited on page 24.)

Eser Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 107–116, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. doi: 10.1145/502512.502530. URL http://doi.acm.org/10.1145/502512.502530. (Cited on page 28.)

Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. Slide rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '08, pages 73–80, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-976-0. doi: 10.1145/1414471.1414487. URL http://doi.acm.org/10.1145/1414471.1414487. (Cited on page 158.)

Thorsten Karrer, Moritz Wittenhagen, and Jan Borchers. Draglocks: Handling temporal ambiguities in direct manipulation video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 623–626, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207764. URL http://doi.acm.org/10.1145/2207676.2207764. (Cited on page 91.)

Daniel A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, January 2002. ISSN 1077-2626. doi: 10.1109/2945.981847. URL http://dx.doi.org/10.1109/2945.981847. (Cited on page 23.)

Daniel A. Keim and Hans-Peter Krigel. Visdb: Database exploration using multidimensional visualization. *IEEE Comput. Graph. Appl.*, 14(5):40–49, September 1994. ISSN 0272-1716. doi: 10.1109/38.310723. URL http://dx.doi.org/10.1109/38.310723. (Cited on page 21.)

Brittany Kondo and Christopher Collins. Dimpvis: Exploring time-varying information visualizations by direct manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 2014. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2346250. (Cited on pages 92, 95, and 107.)

Robert Kosara, Helwig Hauser, and Donna L Gresh. An interaction view on information visualization. *Eurographics 2003 - STAR*, 2003. (Cited on pages 4 and 23.)

Donald L. Kreher and Douglas R. Stinson. Combinatorial algorithms: Generation, enumeration, and search. *SIGACT News*, 30(1):33–35, March 1999. ISSN 0163-5700. doi: 10.1145/309739.309744. URL http://doi.acm.org/10.1145/309739.309744. (Cited on page 68.)

Gordon Kurtenbach and William Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 482–487, New York, NY, USA, 1993. ACM. ISBN 0-89791-575-5. doi: 10.1145/169059.169426. URL http://doi.acm.org/10.1145/169059.169426. (Cited on page 34.)

Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 258–264, New York, NY, USA, 1994. ACM. ISBN 0-89791-650-6. doi: 10.1145/191666.191759. URL http://doi.acm.org/10.1145/191666.191759. (Cited on page 34.)

Ernest Kwan, Irene R. R. Lu, and Michael Friendly. Tableplot: A new tool for assessing precise predictions. *Zeitschrift für Psychologie / Journal of Psychology*, 217(1):38–48, 2009. doi: 10.1027/0044-3409.217.1.38. (Cited on page 121.)

Bum chul Kwon, Waqas Javed, Niklas Elmqvist, and Ji Soo Yi. Direct manipulation through surrogate objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 627–636, New

York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979033. URL http://doi.acm.org/10.1145/1978942.1979033. (Cited on pages 29 and 92.)

Heidi Lam, Tamara Munzner, and Robert Kincaid. Overview use in multiple visual information resolution interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1278–1285, November 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70583. URL http://dx.doi.org/10.1109/TVCG.2007.70583. (Cited on pages 41 and 43.)

Brenda K. Laurel. Interface as mimesis. In D. A. Norman and S. W. Draper, editors, *User Centered System Design: New Perspectives on Human-Computer Interaction*, pages 67–85. Erlbaum, 1986. (Cited on pages ix and 14.)

J. Le Fourn and A. Mailles. Un micro ordinateur graphique : Programmes réalisés et exemples d'application. 1984. URL http://hdl.handle.net/2042/10838. (Cited on page 122.)

Bongshin Lee, Petra Isenberg, Nathalie Henry Riche, and Sheelagh Carpendale. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2689–2698, Dec 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.204. (Cited on pages 4, 6, 11, 21, and 36.)

Nathaniel Leibowitz, Barak Baum, Giora Enden, and Amir Karniel. The exponential learning equation as a function of successful trials results in sigmoid performance. *Journal of Mathematical Psychology*, 54(3):338 – 340, 2010. ISSN 0022-2496. doi: http://dx.doi.org/10.1016/j.jmp.2010.01.006. URL http://www.sciencedirect.com/science/article/pii/S0022249610000179. (Cited on page 159.)

Jing Li, Jean-Bernard Martens, and Jarke J. van Wijk. A model of symbol size discrimination in scatterplots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2553–2562, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753714. URL http://doi.acm.org/10.1145/1753326.1753714. (Cited on page 134.)

Innar Liiv. Seriation and matrix reordering methods: An historical overview. *Stat. Anal. Data Min.*, 3(2):70–91, April 2010. ISSN 1932-1864. doi: 10.1002/sam.v3:2. URL http://dx.doi.org/10.1002/sam.v3:2. (Cited on pages 116, 118, 119, and 144.)

Sharon Lin, Julie Fortuna, Chinmay Kulkarni, Maureen Stone, and Jeffrey Heer. Selecting semantically-resonant colors for data visualization. In *Proceedings of the 15th Eurographics Conference on Visualization*, EuroVis '13, pages 401–410, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics. doi: 10.1111/cgf.12127. URL http://dx.doi.org/10.1111/cgf.12127. (Cited on pages 76 and 143.)

Zhicheng Liu and John Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):999–1008, November 2010. ISSN 1077-2626. doi: 10.1109/TVCG.2010.177. URL http://dx.doi.org/10.1109/TVCG.2010.177. (Cited on pages 4 and 23.)

Andrew Lumsdaine, Chris Weaver, and Joseph A. Cottam. Watch this: A taxonomy for dynamic data visualization. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, VAST '12, pages 193–202, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-1-4673-4752-5. doi: 10.1109/VAST.2012.6400552. URL http://dx.doi.org/10.1109/VAST.2012.6400552. (Cited on page 108.)

Jock Mackinlay. *Automatic Design of Graphical Representations*. PhD thesis, Stanford, California, 1986. (Cited on page 21.)

Theo Mandel. *The Elements of User Interface Design*. John Wiley & Sons, Inc., New York, NY, USA, 1997. ISBN 0-471-16267-1. (Cited on page 19.)

Allen R. Martin and Matthew O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th Conference on Visualization '95*, VIS '95, pages 271–, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-7187-4. URL http://dl.acm.org/citation.cfm?id=832271.833844. (Cited on page 27.)

David McGookin, Euan Robertson, and Stephen Brewster. Clutching at straws: Using tangible interaction to provide non-visual access to graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1715–1724, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753583. URL http://doi.acm.org/10.1145/1753326.1753583. (Cited on page 24.)

M. J. McGuffin, N. Burtnyk, and G. Kurtenbach. Fast sliders: Integrating marking menus and the adjustment of continuous values. In *Proceedings of the Symposium on Graphics Interface*, GI '02, pages 35–41, 2002. (Cited on page 127.)

Michael J. McGuffin and Ravin Balakrishnan. Interactive visualization of genealogical graphs. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 3–, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7803-9464-x. doi: 10.1109/INFOVIS.2005.22. URL http://dx.doi.org/10.1109/INFOVIS.2005.22. (Cited on page 34.)

A. Michotte. *The perception of causality*. Methuen's manuals of modern psychology. Basic Books, 1963. URL http://books.google.fr/books?id=NOt9AAAAMAAJ. (Cited on page 14.)

Tomer Moscovich, Fanny Chevalier, Nathalie Henry Riche, Emmanuel Pietriga, and Jean-Daniel Fekete. Topology-aware navigation in large networks. In *Proc. CHI'09*, pages 2319–2328. ACM, 2009. (Cited on pages 33, 91, and 92.)

Tamara Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, November 2009. ISSN 1077-2626. doi: 10.1109/TVCG.2009.111. URL http://dx.doi.org/10.1109/TVCG.2009.111. (Cited on page 7.)

Allen Newell and Paul S. Rosenbloom. The soar papers (vol. 1). chapter Mechanisms of Skill Acquisition and the Law of Practice, pages 81–135. MIT Press, Cambridge, MA, USA, 1993. ISBN 0-262-68071-8. URL http://dl.acm.org/citation.cfm?id=162580.162586. (Cited on page 159.)

Jakob Nielsen. Noncommand user interfaces. *Commun. ACM*, 36(4):83–99, April 1993. ISSN 0001-0782. doi: 10.1145/255950.153582. URL http://doi.acm.org/10.1145/255950.153582. (Cited on page 5.)

Donald A. Norman. Cognitive engineering. In D. A. Norman and S. W. Draper, editors, *User Centered System Design: New Perspectives on Human-Computer Interaction*, pages 31–61. Erlbaum, Hillsdale, NJ, 1986. (Cited on page 13.)

Donald A. Norman. *The Psychology Of Everyday Things*. Basic Books, 1988. ISBN 0465067093. (Cited on pages 6, 19, and 160.)

Donald A. Norman. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993. ISBN 0-201-62695-0. (Cited on page 19.)

Donald A. Norman and Jakob Nielsen. Gestural interfaces: A step backward in usability. *interactions*, 17(5):46–49, September 2010. ISSN 1072-5520. doi: 10.1145/1836216.1836228. URL http://doi.acm.org/10.1145/1836216.1836228. (Cited on pages 158 and 160.)

Chris North. Toward measuring visualization insight. *IEEE Comput. Graph. Appl.*, 26(3):6–9, May 2006. ISSN 0272-1716. doi: 10.1109/MCG.2006.70. URL http://dx.doi.org/10.1109/MCG.2006.70. (Cited on page 154.)

Chris North, Tim Dwyer, Bongshin Lee, Danyel Fisher, Petra Isenberg, George Robertson, and Kori Inkpen. Understanding multi-touch manipulation for surface computing. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, INTERACT '09, pages 236–249, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-03657-6. doi: 10.1007/978-3-642-03658-3_31. URL http://dx.doi.org/10.1007/978-3-642-03658-3_31. (Cited on page 24.)

Luke Olsen, Faramarz F. Samavati, Mario Costa Sousa, and Joaquim A. Jorge. Technical section: Sketch-based modeling: A survey. *Comput. Graph.*, 33(1):85–103, February 2009. ISSN 0097-8493. doi: 10.1016/j.cag.2008.09.013. URL http://dx.doi.org/10.1016/j.cag.2008.09.013. (Cited on pages 5 and 14.)

P. Orús and G. Villarroya. TGINF : graphic information processing for teaching. Reims, France, 2003. URL http://edutice.archives-ouvertes.fr/edutice-00001366. (Cited on page 122.)

Gilles Palsky. *L'esprit des cartes: approches historiques, sémiologiques et sociologiques en cartographie: diplôme d'habilitation à diriger des recherches*. Université de Paris XII-Val-de-Marne, 2003. (Cited on pages 114, 115, and 116.)

Lewis. Peterson and Wilbur Schramm. How accurately are different kinds of graphs read? *Audiovisual communication review*, 2(3):178–189, 1954. ISSN 0001-2890. doi: 10.1007/BF02713334. URL http://dx.doi.org/10.1007/BF02713334. (Cited on page 41.)

Michael P. Peterson. Active legends for interactive cartographic animation. *International Journal of Geographical Information Science*, 13(4):375–383, 1999. doi: 10.1080/136588199241256. URL http://dx.doi.org/10.1080/136588199241256. (Cited on page 29.)

William A. Pike, John Stasko, Remco Chang, and Theresa A. O'Connell. The science of interaction. *Information Visualization*, 8(4):263–274, December 2009. ISSN 1473-8716. doi: 10.1057/ivs.2009.22. URL http://dx.doi.org/10.1057/ivs.2009.22. (Cited on pages 4, 15, and 153.)

W. Playfair, H. Wainer, and I. Spence. *Playfair's commercial and political atlas and statistical breviary*. Cambridge University Press, 2005. ISBN 9780521855549. (Cited on pages 66 and 69.)

William Playfair. *The Commercial and Political Atlas*. J. Debrett, London, 1st edition, 1786. (Cited on pages 2 and 41.)

Karl Raimund Popper. *The logic of scientific discovery*. Hutchinson London, 1959. (Cited on page 37.)

Alexander J. Quinn and Benjamin B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 1403–1412. ACM, 2011. (Cited on page 77.)

Ramana Rao and Stuart K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 318–322, New York, NY, USA, 1994. ACM. ISBN 0-89791-650-6. doi: 10.1145/191666.191776. URL http://doi.acm.org/10.1145/191666.191776. (Cited on pages 66, 68, and 121.)

Hannes Reijner. The development of the horizon graph. available online at http://www.stonesc.com/Vis08_Workshop/DVD/Reijner_submission.pdf, 2008. (Cited on page 41.)

George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, November 2008. ISSN 1077-2626. doi: 10.1109/TVCG.2008.125. URL http://dx.doi.org/10.1109/TVCG.2008.125. (Cited on page 91.)

W. S. Robinson. A method for chronologically ordering archaeological deposits. *American Antiquity*, 16(4):293–301, 1951. (Cited on page 119.)

Hans Rosling. Gapminder. *GapMinder Foundation http://www. gapminder. org*, 2009. (Cited on pages 91, 97, and 105.)

Steve Rubin. Available at www.eecs.berkeley.edu/~srubin/toys/reorderable/. (Cited on pages 122, 123, and 133.)

Kathy Ryall, Neal Lesh, Tom Lanning, Darren Leigh, Hiroaki Miyashita, and Shigeru Makino. Querylines: Approximate query for visual browsing. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1765–1768, New York, NY, USA, 2005. ACM. ISBN 1-59593-002-7. doi: 10.1145/1056808.1057017. URL http://doi.acm.org/10.1145/1056808.1057017. (Cited on page 24.)

Jeffrey M. Rzeszotarski and Aniket Kittur. Touchviz: (multi)touching multivariate data. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 3119–3122, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1952-2. doi: 10.1145/2468356.2479625. URL http://doi.acm.org/10.1145/2468356.2479625. (Cited on page 30.)

Jeffrey M. Rzeszotarski and Aniket Kittur. Kinetica: Naturalistic multi-touch data visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 897–906, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557231. URL http://doi.acm.org/10.1145/2556288.2557231. (Cited on page 30.)

Takafumi Saito, Hiroko Nakamura Miyamura, Mitsuyoshi Yamamoto, Hiroki Saito, Yuka Hoshiya, and Takumi Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 23–, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7803-9464-x. doi: 10.1109/INFOVIS.2005.35. URL http://dx.doi.org/10.1109/INFOVIS.2005.35. (Cited on page 41.)

Günther Sawitzki. Extensible statistical software: On a voyage to oberon. *Journal of Computational and Graphical Statistics*, 5(3):263–283, 1996. doi: 10.1080/10618600.1996.10474711. (Cited on pages 26, 114, 121, 123, and 133.)

Günther Sawitzki. Bertin matrices. an r implementation, 2012. URL http://bertin.r-forge.r-project.org/bertinR.pdf. (Cited on page 123.)

Sebastian Schmidt, Miguel A. Nacenta, Raimund Dachselt, and Sheelagh Carpendale. A set of multi-touch graph interaction techniques. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 113–116, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0399-6. doi: 10.1145/1936652.1936673. URL http://doi.acm.org/10.1145/1936652.1936673. (Cited on page 24.)

Ayelet Segal. *Do gestural interfaces promote thinking? Embodied interaction: Congruent gestures and direct touch promote performance in math*. PhD thesis, COLUMBIA UNIVERSITY, 2011. (Cited on page 16.)

Ayelet Segal, Barbara Tversky, and John Black. Conceptually congruent actions can promote thought. *Journal of Applied Research in Memory and Cognition*, (0), 2014. ISSN 2211-3681. doi: http://dx.doi.org/10.1016/j.jarmac.2014.06.004. URL http://www.sciencedirect.com/science/article/pii/S2211368114000552. (Cited on page 16.)

Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2): 96–113, July 2005. ISSN 1473-8716. doi: 10.1057/palgrave.ivs.9500091. URL http://dx.doi.org/10.1057/palgrave.ivs.9500091. (Cited on page 68.)

Mindy Seto, Stacey Scott, and Mark Hancock. Investigating menu discoverability on a digital tabletop in a public setting. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 71–80, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396647. URL http://doi.acm.org/10.1145/2396636.2396647. (Cited on page 160.)

Conglei Shi, Weiwei Cui, Shixia Liu, Panpan Xu, Wei Chen, and Huamin Qu. Rankexplorer: Visualization of ranking changes in large time series data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2669–2678, Dec 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.253. (Cited on pages 66 and 69.)

B. Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8):57–69, August 1983. ISSN 0018-9162. doi: 10.1109/MC.1983.1654471. URL http://dx.doi.org/10.1109/MC.1983.1654471. (Cited on pages 5, 12, 13, 17, 18, 75, 122, and 158.)

Ben Shneiderman. The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology*, 1(3):237–256, 1982. doi: 10.1080/01449298208914450. (Cited on page 12.)

Ben Shneiderman. Dynamic queries for visual information seeking. *IEEE Softw.*, 11(6):70–77, November 1994. ISSN 0740-7459. doi: 10.1109/52.329404. URL http://dx.doi.org/10.1109/52.329404. (Cited on pages 25 and 73.)

Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, pages 336–, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7508-X. URL http://dl.acm.org/citation.cfm?id=832277.834354. (Cited on pages 4, 23, and 92.)

Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *interactions*, 4(6):42–61, November 1997. ISSN 1072-5520. doi: 10.1145/267505.267514. URL http://doi.acm.org/10.1145/267505.267514. (Cited on page 20.)

Ben Shneiderman and Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Pearson Addison Wesley, 2004. ISBN 0321197860. (Cited on pages 12 and 13.)

H. Siirtola. Interaction with the reorderable matrix. In *Information Visualization, 1999. Proceedings. 1999 IEEE International Conference on*, pages 272–277, 1999. doi: 10.1109/IV.1999.781570. (Cited on pages 26, 114, 120, 121, 122, and 133.)

Harri Siirtola. Interactive cluster analysis. In *Proceedings of the Information Visualisation, Eighth International Conference*, IV '04, pages 471–476, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2177-0. doi: 10.1109/IV.2004.84. URL http://dx.doi.org/10.1109/IV.2004.84. (Cited on page 133.)

Harri Siirtola and Erkki Mäkinen. Constructing and reconstructing the reorderable matrix. *Information Visualization*, 4(1):32–48, March 2005. ISSN 1473-8716. doi: 10.1057/palgrave.ivs.9500086. URL http://dx.doi.org/10.1057/palgrave.ivs.9500086. (Cited on pages 118, 119, 122, and 133.)

Harri Siirtola and Kari-Jouko Räihä. Discussion: Interacting with parallel coordinates. *Interact. Comput.*, 18(6):1278–1309, December 2006. ISSN 0953-5438. doi: 10.1016/j.intcom.2006.03.006. URL http://dx.doi.org/10.1016/j.intcom.2006.03.006. (Cited on page 27.)

Sônia Fernandes Silva and Tiziana Catarci. Visualization of linear time-oriented data: A survey. In *Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00)-Volume 1 - Volume 1*, WISE '00, pages 310–, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0577-5-1. URL http://dl.acm.org/citation.cfm?id=882511.885379. (Cited on page 41.)

David Simkin and Reid Hastie. An Information-Processing Analysis of Graph Perception. *Journal of the American Statistical Association*, 82(398): 454–465, 1987. ISSN 01621459. (Cited on pages 41, 44, and 50.)

David Canfield Smith, Charles Irby, Ralph Kimball, and Eric Harslem. The star user interface: An overview. In *Proceedings of the June 7-10, 1982, National Computer Conference*, AFIPS '82, pages 515–528, New York, NY, USA, 1982. ACM. ISBN 0-88283-035-X. doi: 10.1145/1500774.1500840. URL http://doi.acm.org/10.1145/1500774.1500840. (Cited on pages 5 and 12.)

Randall B. Smith. Experiences with the alternate reality kit: An example of the tension between literalism and magic. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, pages 61–67, New York, NY, USA, 1987. ACM. ISBN 0-89791-213-6. doi: 10.1145/29933.30861. URL http://doi.acm.org/10.1145/29933.30861. (Cited on page 20.)

Awalin Sopan, Manuel Freire, Catherine Plaisant, Jennifer Golbeck, and Ben Shneiderman. Distribution column overviews in tabular visualizations. Technical report, HCIL, 2010. (Cited on page 68.)

R. William Soukoreff and I. Scott MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *Int. J. Hum.-Comput. Stud.*, 61(6):751–789, December 2004. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2004.09.001. URL http://dx.doi.org/10.1016/j.ijhcs.2004.09.001. (Cited on page 161.)

Robert Spence. *Information Visualization: Design for Interaction (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007. ISBN 0132065509. (Cited on pages 23 and 24.)

Michael Spenke, Christian Beilken, and Thomas Berlage. Focus: The interactive table for product comparison and selection. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pages 41–50, New York, NY, USA, 1996. ACM. ISBN 0-89791-798-7. doi: 10.1145/237091.237097. URL http://doi.acm.org/10.1145/237091.237097. (Cited on page 121.)

John T. Stasko. Title: The value of visualization...and why interaction matters. available online at http://vimeo.com/98986594, 2014. (Cited on pages 154 and 163.)

Edward A Suchman. The scalogram board technique for scale analysis. *Studies in Social Psychology in World War II*, 4:91–121, 1950. (Cited on page 116.)

Yiwen Sun, Jason Leigh, Andrew Johnson, and Sangyoon Lee. Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *Proceedings of the 10th International Conference on Smart Graphics*, SG'10, pages 184–195, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-13543-9, 978-3-642-13543-9. URL http://dl.acm.org/citation.cfm?id=1894345.1894369. (Cited on page 24.)

Tableau Software, 2014. www.tableausoftware.com/. (Cited on pages 21, 120, and 162.)

J. Talbot, J. Gerth, and P. Hanrahan. An empirical model of slope ratio comparisons. *IEEE Transactions on Visualization and Computer Graphics*, 18(12): 2613–2620, Dec 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.196. URL http://vis.stanford.edu/papers/slope-ratio-comparison. (Cited on page 43.)

James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE CS Press, 2005. ISBN 0769523234. (Cited on pages 4 and 23.)

Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale. Lark: Coordinating co-located collaboration with information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1065–1072, Nov 2009. ISSN 1077-2626. doi: 10.1109/TVCG.2009.162. (Cited on page 23.)

Anne Treisman. Preattentive processing in vision. *Comput. Vision Graph. Image Process.*, 31(2):156–177, August 1985. ISSN 0734-189X. doi: 10.1016/S0734-189X(85)80004-9. URL http://dx.doi.org/10.1016/S0734-189X(85)80004-9. (Cited on page 2.)

Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986. ISBN 0-9613921-0-X. (Cited on pages 2, 29, 41, 66, 68, 69, 73, and 95.)

Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. Animation: Can it facilitate? *Int. J. Hum.-Comput. Stud.*, 57(4):247–262, October 2002. ISSN 1071-5819. doi: 10.1006/ijhc.2002.1017. URL http://dx.doi.org/10.1006/ijhc.2002.1017. (Cited on pages ix and 16.)

Lisa Tweedie. Characterizing interactive externalizations. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 375–382, New York, NY, USA, 1997. ACM. ISBN 0-89791-802-9. doi: 10.1145/258549.258803. URL http://doi.acm.org/10.1145/258549.258803. (Cited on pages 21, 24, 28, and 36.)

Lisa Tweedie, Bob Spence, David Williams, and Ravinder Bhogal. The attribute explorer. In *Conference Companion on Human Factors in Computing Systems*, CHI '94, pages 435–436, New York, NY, USA, 1994. ACM. ISBN 0-89791-651-4. doi: 10.1145/259963.260433. URL http://doi.acm.org/10.1145/259963.260433. (Cited on page 30.)

Andries Van Dam. Post-wimp user interfaces. *Commun. ACM*, 40(2):63–67, February 1997. ISSN 0001-0782. doi: 10.1145/253671.253708. URL http://doi.acm.org/10.1145/253671.253708. (Cited on pages 5, 7, 11, 12, 14, 19, 20, and 21.)

Jarke J. van Wijk. The value of visualization. In *Visualization, 2005. VIS 05. IEEE*, pages 79–86, Oct 2005. doi: 10.1109/VISUAL.2005.1532781. (Cited on pages 6, 7, 35, and 160.)

Françoise Vergneault, Roland Lamontagne, and Jacques Bertin. Traitement graphique d'une information: les marines royales de france et de grande-bretagne (1697-1747). *Annales. Économies, Sociétés, Civilisations*, 22(5):991–1004, 1967. ISSN 0395-2649. doi: 10.3406/ahess.1967.421596. (Cited on pages 116 and 131.)

Bret Victor. Drawing dynamic visualizations. Stanford HCI Seminar. Video available at http://vimeo.com/66085662, 2013a. (Cited on page 143.)

Bret Victor. Tangle: a javascript library for reactive documents, 2013b. http://worrydream.com/Tangle/. (Cited on pages 75 and 85.)

Luc Vlaming, Christopher Collins, Mark Hancock, Miguel Nacenta, Tobias Isenberg, and Sheelagh Carpendale. Integrating 2d mouse emulation with 3d manipulation for visualizations on a multi-touch table. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 221–230, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0399-6. doi: 10.1145/1936652.1936693. URL http://doi.acm.org/10.1145/1936652.1936693. (Cited on page 24.)

Stephen Voida, Matthew Tobiasz, Julie Stromer, Petra Isenberg, and Sheelagh Carpendale. Getting practical with interactive tabletop displays: Designing for dense data, "fat fingers," diverse interactions, and face-to-face collaboration. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 109–116, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731926. URL http://doi.acm.org/10.1145/1731903.1731926. (Cited on page 24.)

Matthew O. Ward and Jing Yang. Interaction spaces in data and information visualization. In *Proceedings of the Sixth Joint Eurographics - IEEE TCVG Conference on Visualization*, VISSYM'04, pages 137–146, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics. ISBN 3-905673-07-X. doi: 10.2312/VisSym/VisSym04/137-146. URL http://dx.doi.org/10.2312/VisSym/VisSym04/137-146. (Cited on page 24.)

Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. ISBN 1558608192. (Cited on page 2.)

Martin Wattenberg. Sketching a graph to query a time-series database. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, pages 381–382, New York, NY, USA, 2001. ACM. ISBN 1-58113-340-5. doi: 10.1145/634067.634292. URL http://doi.acm.org/10.1145/634067.634292. (Cited on page 24.)

Martin Wattenberg. A note on space-filling visualizations and space-filling curves. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 24–, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7803-9464-x. doi: 10.1109/INFOVIS.2005.1. URL http://dx.doi.org/10.1109/INFOVIS.2005.1. (Cited on page 41.)

Peter Wegner. Why interaction is more powerful than algorithms. *Commun. ACM*, 40(5):80–91, May 1997. ISSN 0001-0782. doi: 10.1145/253769.253801. URL http://doi.acm.org/10.1145/253769.253801. (Cited on pages 4 and 118.)

Stephen Wehrend and Clayton Lewis. A problem-oriented classification of visualization techniques. In *Proceedings of the 1st Conference on Visualization '90*, VIS '90, pages 139–143, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press. ISBN 0-8186-2083-8. URL http://dl.acm.org/citation.cfm?id=949531.949553. (Cited on pages 4 and 23.)

Steve Whittaker. Next generation interfaces. In *Proc. of the AAAI Symposium on Knowledge-Based Human-Commputer Communication*, pages 127–131, Stanford, CA, 1990. (Cited on pages 18 and 19.)

Wikipedia. Pivot table, 2013. http://en.wikipedia.org/wiki/Pivot_table. (Cited on page 68.)

Leland Wilkinson. Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist*, 54(8):594–604, August 1999. (Cited on pages 116 and 118.)

Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387245448. (Cited on page 23.)

Leland Wilkinson and Michael Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009. doi: 10.1198/tas.2009.0033. (Cited on pages 114 and 116.)

Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, Nov 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70589. (Cited on page 92.)

Christopher Williamson and Ben Shneiderman. The dynamic homefinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 338–346, New York, NY, USA, 1992. ACM. ISBN 0-89791-523-2. doi: 10.1145/133160.133216. URL http://doi.acm.org/10.1145/133160.133216. (Cited on page 25.)

Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1083–1092, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518866. URL http://doi.acm.org/10.1145/1518701.1518866. (Cited on pages 5 and 14.)

M. Wolter, B. Hentschel, Irene Tedjo-Palczynski, and T. Kuhlen. A direct manipulation interface for time navigation in scientific visualizations. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*, pages 11–18, March 2009. doi: 10.1109/3DUI.2009.4811199. (Cited on pages 91 and 92.)

Han-Ming Wu, Yin-Jing Tien, and Chun-houh Chen. Gap: A graphical environment for matrix visualization and cluster analysis. *Computational Statistics & Data Analysis*, 54(3):767–778, March 2010. ISSN 0167-9473. doi: 10.1016/j.csda.2008.09.029. (Cited on page 123.)

Ji Soo Yi, Rachel Melton, John Stasko, and Julie A. Jacko. Dust & magnet: Multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, October 2005. ISSN 1473-8716. doi: 10.1057/palgrave.ivs.9500099. URL http://dx.doi.org/10.1057/palgrave.ivs.9500099. (Cited on page 30.)

Ji Soo Yi, Youn ah Kang, John Stasko, and Julie Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, November 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70515. URL http://dx.doi.org/10.1109/TVCG.2007.70515. (Cited on pages 4, 23, and 92.)

Jian Zhao, Fanny Chevalier, and Ravin Balakrishnan. Kronominer: Using multi-foci navigation for the visual exploration of time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1737–1746, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979195. URL http://doi.acm.org/10.1145/1978942.1979195. (Cited on pages 19 and 31.)

Michelle X. Zhou and Steven K. Feiner. Visual task characterization for automated visual discourse synthesis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 392–399, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-30987-4. doi: 10.1145/274644.274698. URL http://dx.doi.org/10.1145/274644.274698. (Cited on page 24.)